VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

NÁVRH A OVĚŘENÍ FUNKČNOSTI SYSTÉMU SMĚROVÉHO ŘÍZENÍ VOZIDLA

DESIGN AND EVALUATION OF VEHICLE STEERING CONTROLLER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — **Bc. MARGETAJ MARTIN**

AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. ROBERT GREPL, Ph.D.**

SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání diplomové práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	**Bc. Martin Margetaj**
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Mechatronika
Vedoucí práce:	**doc. Ing. Robert Grepl, Ph.D.**
Akademický rok:	2018/19

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Návrh a ověření funkčnosti systému směrového řízení vozidla

Stručná charakteristika problematiky úkolu:

Moderní pokročilé asistenční systémy ve vozidlech jsou schopné nahradit člověka v čím dál větším počtu činností souvisejících s řízením vozidla. Jako příklad můžeme uvést adaptivní tempomat, automatické udržování vozidla v pruhu nebo automatický parkovací systém. Systémy této kategorie se nadále prudce rozvíjí s cílem dosažení plně autonomního vozidla. Pro správnou funkci těchto vysokoúrovňových systémů jsou třeba spolehlivé a robustní řídicí systémy pro ovládání základních mechanických skupin vozidla. Tato diplomová práce se zabývá návrhem a vyhodnocením funkčnosti systému směrového řízení vozidla.

Cíle diplomové práce:

1. Seznamte se se základním principem zatáčení vozidla a se základním principem posilovače řízení, zaměřte se především na elektromechanický posilovač řízení.
2. Seznamte se se síťovými protokoly CAN a FlexRay.
3. Seznamte se s infrastrukturou testovacího vozidla potřebnou pro ovládání posilovače řízení, identifikujte FlexRay zprávy pro ovládání posilovače řízení.
4. Navrhněte systém směrového řízení vozidla.
5. Navrhněte nadřazenou logiku, která bude generovat referenční hodnotu pro systém směrového řízení vozidla, referenční hodnota bude taková, aby vozidlo projelo trať tzv. losího testu.
6. Vyhodnoťte funkčnost řídicího systému v losím testu, cílem je projekt losí test minimálně v rychlosti 50 km/h.

Seznam doporučené literatury:

JAZAR, Reza N., Vehicle Dynamics, Springer, 2017

KIENCKE, U., NIELSEN, L., Automotive Control Systems, Springer, 2007

SMITH, C., The Car Hacker's Handbook, No Starch Press, 2017

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc. ředitel ústavu	doc. Ing. Jaroslav Katolický, Ph.D. děkan fakulty

Abstrakt
V této práci je prezentován a popsán FlexRay komunikační protokol se sběrnicí. Je představeno několik matematických modelů vozidel pro simulaci a pro vývoj řídících systémů. Dále je popsána tvorba signálové brány pro testovací vozidlo. Software pro úpravu signálu je implementován a odzkoušen v testovacím vozidle. Matematick0 modely byli parametrizovány aby odpovídali skutečnému vozidlu. Vyvinuté řídící systémy pro ovládání vozidla skrze losí test byli implementovány do automobilu a jejich vlastnosti byli otestovány. Závěrem jsou prezentovány výsledky testů jednotlivých řídících systémů.

Abstract
FlexRay communication protocol and bus are presented and discussed in this thesis. Several vehicle mathematical models of the vehicle dynamic for simulations and development of the control system are introduced. Further, the design of a signal gateway is presented. Software for editing vehicle signals is introduced and tested on the test vehicle. Vehicle models were parametrized to match real test vehicle behavior. Developed control systems were implemented to the test vehicle, and their performance is tested, the results of all tests are compared and evaluated.

Klíčová slova
EPS, losí test, FlexRay protokol, dynamika vozidla, Vector, VN8914, VN7600, software.

Keywords
EPS, double lane change, FlexRay protocol, vehicle dynamics, Vector, VN8914, VN7600, software

MARGETAJ, M. *Návrh a ověření funkčnosti systému směrového řízení vozidla.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2019. 78 s. Vedoucí diplomové práce doc. Ing. Robert Grepl, Ph.D..

I declare that I have written the Master's Thesis titled "Návrh a ověření funkčnosti systému směrového řízení vozidla" independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the thesis and listed in the comprehensive bibliography at the end of the thesis. As the author I furthermore declare that, with respect to the creation of this Master's Thesis, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation S 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll., Section 2, Head VI, Part 4.

Bc. Margetaj Martin

I would like to thank my supervisor doc. Ing. Robert Grepl, Ph.D. for his valuable advice and guidance during the creation of this thesis.

Great thanks also belong to all employees and representatives of Porsche Engineering Services s.r.o. for every helpful consultations and for providing the experimental vehicle. Special thanks goes to all the test drivers, for their willingness to participate in experimental test.

I also thank my parents, friends and girlfriend for support, without which this work would not be completed.

Bc. Margetaj Martin

Contents

1 Introduction **13**

2 Background research **15**
2.1 Electric power steering . . . 15
2.1.1 Principle of EPS . . . 15
2.2 Introduction to the vehicle communication . . . 16
2.2.1 CAN . . . 16
2.2.2 FlexRay . . . 17
2.3 Introduction to the vehicle modeling . . . 26
2.3.1 Tire model . . . 26
2.3.2 Kinematic model . . . 27
2.3.3 Dynamic single track model . . . 28

3 Goals and design of thesis **33**

4 Design and implementation **37**
4.1 Design of signal gateway . . . 37
4.1.1 Lane keeping assistant . . . 37
4.1.2 Modified signals . . . 40
4.2 Design of PID regulator . . . 42
4.2.1 PID regulator scheme . . . 43
4.2.2 Setting constants . . . 45
4.3 the vehicle parametrization . . . 46
4.3.1 Test vehicle . . . 46
4.4 Parameter validation . . . 51
4.5 The vehicle simulation . . . 53
4.5.1 Simulink model . . . 53
4.5.2 Test manoeuvre . . . 55
4.5.3 Simulations . . . 56
4.6 Experimental models . . . 59
4.6.1 Feedback control model . . . 59
4.6.2 Feedforward control model . . . 61
4.6.3 Feedforward and feedback control model . . . 63

5 Experimental results **65**
5.1 Position feedback . . . 65
5.1.1 Results . . . 65
5.2 Feedforward . . . 65
5.2.1 Results . . . 66
5.3 Feedforward and position feedback . . . 66
5.3.1 Low speed results . . . 66
5.3.2 High speed results . . . 67
5.3.3 Yaw rate feedback . . . 68
5.4 Evaluation of the tests . . . 68

6 Conclusion and future work **69**

References **71**

List of abbreviations **73**

List of symbols **75**

Content of CD **77**

1. Introduction

The latest spread of the autonomous vehicles in the vehicle industry promises to bring a reduction in car accidents by reducing the human contribution to driving and thus reducing the human error. Before self-driving vehicles become widespread, car companies and scientists can use them to make standardized tests to determine quality and safety of the vehicle.Nowadays these tests depends on humans to make the same exact sequence of inputs to the test vehicle, which is nearly impossible.

The objective of this thesis is to design and verify a steering controller that will be able to drive through the testing maneuver without help of the driver. Several control concepts are introduced and experimentally verified on the test vehicle and by the test driver provided by Porsche Engineering Services s.r.o.The communication gateway and the communication read/write system are needed to be designed because the vehicle is not dedicated only for this thesis, which means that the normal function of the vehicle has to be maintained.

For the purposes of this thesis Porsche Engineering Services s.r.o provided the test vehicle, test driver, Vector hardware and FIBEX (described in the section 2.2.2). Basic hardware (PC, cabling, power supply for Vector hardware and differential GPS, etc.) was provided by the PES s.r.o, too. Differential GPS was borrowed from the Czech Institute of Informatics, Robotics, and Cybernetics.

The steering gateway, which is developed in thesis, will be implemented in the test vehicle for future usage of the PES s.r.o, and the PID controller of the front wheel steering angle will be used for the autonomous vehicle driving system, which is being developed by the Czech Institute of Informatics, Robotics, and Cybernetics.

At first, the design, function and behavior of the electric power steering are introduced in the chapter 2.1. Then the communication technologies used in the testing vehicle are introduced and explained in the chapter 2.2. In the chapter 2.3 the mathematical vehicle dynamic model is presented. The summary of the design of the controller is in the chapter 3. Next, the communication gateway design is introduced in the chapter 4.1 and the design of the PID steering controller is in the chapter 4.2. The parametrization of vehicle parameters is in the chapter 4.3 together with the evaluation of the mathematical vehicle model. In the chapter 4.5 the vehicle simulations are made. The experimental results of the presented steering controller are shown in the chapter 5. And finally, conclusion and future work are presented in the chapter 6.

2. Background research

This chapter is dedicated for overview of important technology and principles that are used in thesis.

2.1. Electric power steering

The power steering is a device that allows the driver to turn the steering wheel with ease. There are two types of the power steer, the first is a hydraulic power steer, which lowers the moment that has to be applied on the steering wheel by a hydraulic cylinder, which is part of a servo system. The second is an electrical power steering, which applies moment on the steering rack by an electric motor. This section is focused mainly on the electric power steering as it is the type that is installed in the testing vehicle. More information about the hydraulic power steering can be found in [5].

2.1.1. Principle of EPS

The electric power steering (EPS) is using an electric motor as a source of the moment and assist the driver with steering a vehicle [4]. The force which has to be applied on the steering rack is calculated in a computer model in the vehicle. Inputs for these calculations can as follows, the velocity of the vehicle,the moment on the steering wheel, the position and angular acceleration of the steering wheel. Information about a speed of the vehicle is needed to adjust the moment produced at a higher speed, the vehicle in a higher speed has to react slowly on the inputs and have a dead zone. This has to be precisely determined because, on the other hand the rotating front wheel has a higher gyroscopic moment, which means that it is more difficult to turn them.

The next important property of the power steering is the speed of reactions on the change made by the driver. This can be fatal as the driver has to not only steer the vehicle without power steering but he has to overcome the force of a motor as well.

For safety features, the steering wheel and the steering rack have to be connected mechanically in every vehicle, which means when the power steering stops working for some reason, the driver will has a car still under control, but he has to use same force as no power steering was installed.

In the figure 2.1 we can see a scheme of the electric power steering.

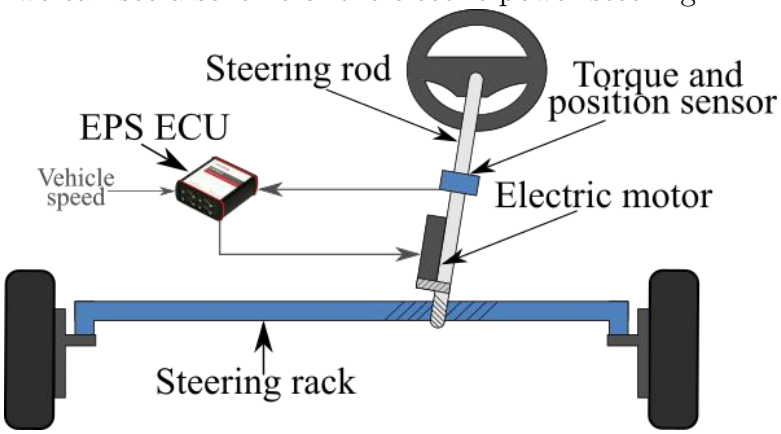


Figure 2.1: Scheme of electric power steering system.

2.2. Introduction to the vehicle communication

The communication within a car is done in three main ways nowadays. In this chapter I look closely on two of these communication buses. In the table 2.1 we can see the comparison of these three communication protocols [8].

BUS	**LIN**	**CAN**	**FlexRay**
Speed	40 kbit/s	1 Mbit/s	10 Mbit/s
Cost	$	$$	$$$
Wires	1	2	2 or 4
Typical Applications	Body Electronics (Mirrors, Power Seats, Accessories)	Powertrain (Engine, Transmission, ABS)	High-Performance Powertrain, Safety (Drive-by-wire, active suspension, adaptive cruise control)

Table 2.1: Comparison of three types of networks [8]

2.2.1. CAN

In this part, I explore CAN communication protocol and the CAN bus. However, this part is not too detailed as everyone knows at least some basic information about the CAN bus, more information can be found in [2].

Basics

The Controller area network (CAN) is a multi-master serial bus standard for connecting nodes (Electronic Control Units). Two or more nodes are mandatory for the CAN network to work. Nodes can be simple as I/O device or complex as the embedded computer with the CAN interface. A general-purpose computer can use these nodes as a gateway to communicate via a USB or Ethernet. All the nodes need to be connected through the two-wire bus. These wires need to be terminated with 120-ohm resistors to work properly [2].

Message transfer

The communication on CAN consists of four different frame types.

- Data Frame - transfers data from the transmitter to receivers.
- Remote Frame - requests Data Frame with the same identifier.
- Error Frame - transmitted whenever any unit detects an error.
- Overload Frame - provides delay after Data or Remote Frame.

In the following part, I focus mostly on the Data Frame as it is used as a primary source of communication with the FlexRay node.

Data frame

As we can see in the figure 2.2, the Data Frame consists of seven-bit fields.

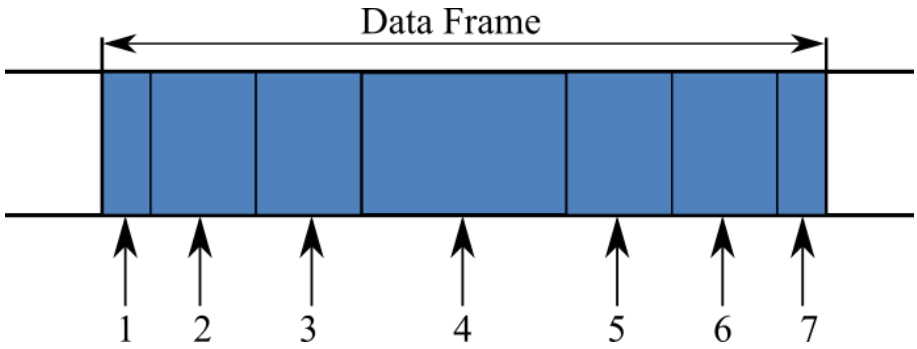


Figure 2.2: CAN data frame.

1. Start of Frame - Also known as SOF marks the beginning of the data frame. Single dominant bit.
2. Arbitration Field - Consists of the identifier and the RTR-BIT.
   - Identifier - 11 bits, and it's transmitted from most significant bits to least.
   - RTR-BIT - Used only when the frame is the REMOTE type frame.
3. Control Field - 6 bits, include the data length code.
4. Data Field - Contains data from 0 to 8 bytes.
5. CRC Field - Contains CRC sequence and CRC delimiter.
   - CRC sequence - Frame check sequence.
   - CRC delimiter - Single recessive bit.
6. ACK Field - All nodes that received correct CRC
7. End of Frame - Marks the end of a data frame. Consist of seven recessive bits.

2.2.2. FlexRay

In this part, I look closely at the FlexRay communication bus. The biggest difference between CAN and Flex-ray bus is, that Flax-ray is a deterministic system, which means that we precisely know when to expect what message from what control unit [8].

Basics

The FlexRay uses the unshielded pair cabling for connecting nodes and it is mainly designed for the dual-channel configuration, which consists of two pairs of wires. The architecture of the dual-cabling is used mainly to reduce external noise on the network without usage of expensive shielding. Most of FlexRay nodes typically have a ground and power wires to power transceivers and microprocessors. Using of the dual-channel configuration opens up new possibilities of enhanced fault-tolerance or increased bandwidth.

The most first-generation FlexRay networks use only one channel to keep costs down.

FlexRay buses require a termination at the ends, in forms of resistors connected between the pair of signal wires, which is the same as with CAN networks. When we are connecting a node to test set-up, we need to remember to use terminating resistors, because a lack of these terminators is the most common problem.

Multi-drop bus topology

Nowadays, the most used topology is the multi-drop bus 2.3 because of its similarities with CAN and LIN bus networks. Each ECU is connected through a short "branch" to the central core (trunk). There are terminating resistors at the ends of wires to eliminate problems with signal reflections. The multi-drop topology perfectly fits with needs of automotive industry by simplifying installation and reducing thorough wiring vehicle.

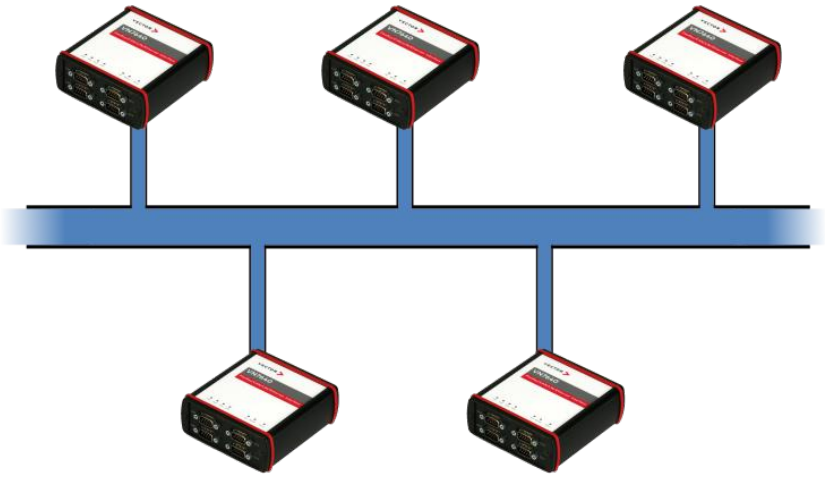


Figure 2.3: Multi-drop bus topology.

Star network topology

The FlexRay standard also supports the star configuration 2.4, which consists of individual links that connect to the central active node. The central node function works as a HUB in PC Ethernet networks. The active star configuration is used for running communications over longer distances or when there is more reliability needed when a part of the system fails. If one part fails the other legs will continue their work. And it can increase the noise immunity by dividing longer wire.

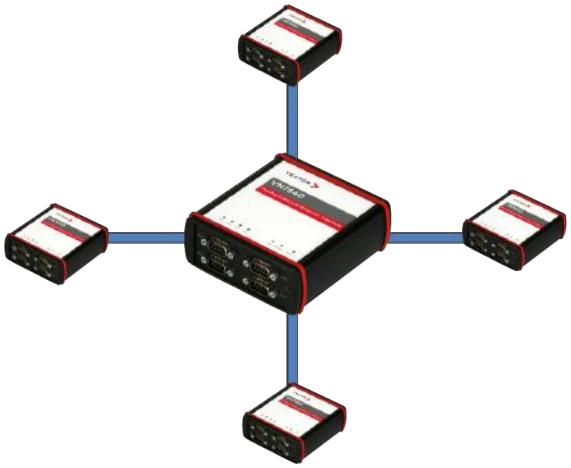


Figure 2.4: Star network topology.

Hybrid network topology

The two methods can be combined to form the hybrid network topology 2.5, which combines performance and reliability of the star network where needed and a low cost and an easy installation of multi-drop topology elsewhere.

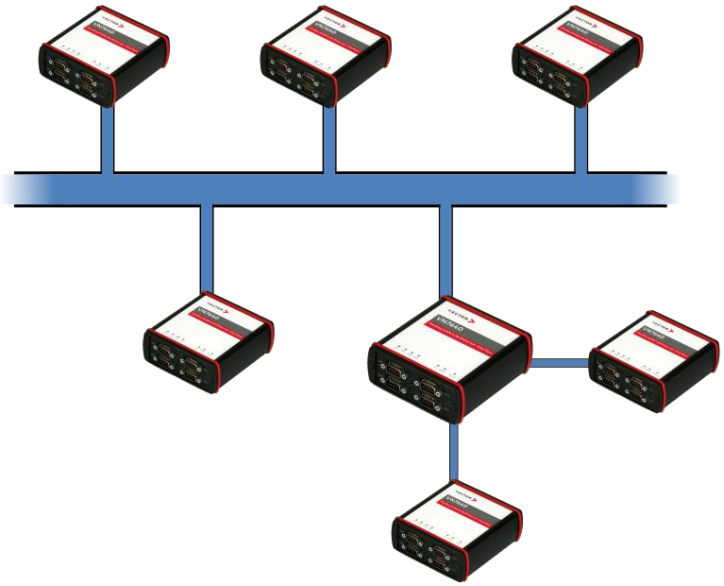


Figure 2.5: Hybrid network topology.

The FlexRay protocol

The FlexRay protocol is a time-triggered protocol that is able to send deterministic data that arrives at a predictable time as well as the CAN-like dynamic event-driven data. This is accomplished by a hybrid communication cycle, which provides a space for static and dynamic data. The cycle is divided into two parts, first is the static part, where deterministic data are sent, and second is the dynamic part, where dynamic data are sent. The sizes of these parts are predefined by the user. To communicate, all nodes of the network must know how the system is configured.

As with any multi-drop bus only one node can write data to the bus at the time. If two nodes were writing at the same time, we would end up with corrupted data. The FlexRay manages nodes with TDMA scheme (Time Division Multiple Access). This means that every node is synchronized according the same clock and every node waits for its turn to send data to the bus. I discuss the main parts of the FlexRay protocol in the following chapters.

The communication cycle

The FlexRay communication cycle is a fundamental element of the media-access scheme within the FlexRay. A duration of the cycle varies from 5 ms to 1 ms and it is fixed. In figure 2.6 we can see individual parts of the cycle. The smallest unit of time on the FlexRay is a macro-tick. And every segment of the cycle consists of many of these macro-ticks.

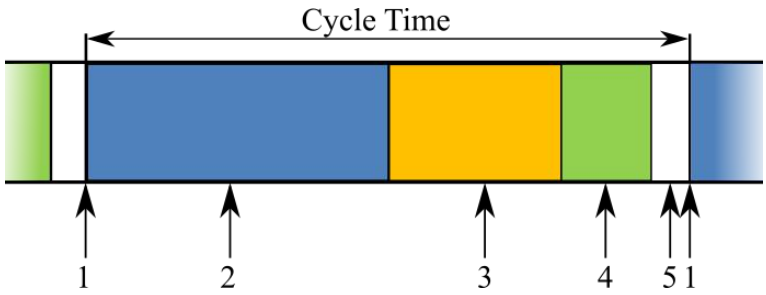


Figure 2.6: FlexRay communication cycle.

1. Cycle Start - Start of a new cycle
2. Static Segment - Reserved slots for deterministic data with a fixed period
3. Dynamic Segment - This part behaves similarly as the CAN, used for event-based data.
4. Symbol Window - Used for the network maintenance and signaling of a starting of the network.
5. Network Idle Time - Used for synchronization maintenance between node clocks.

Static Segment

The blue section in the figure 2.7 is the static segment and it is dedicated to scheduled time-triggered data. The segment is divided into frames, and each frame contains a space reserved for data. When the frame occurs in time, the reserved ECU has the opportunity to transmit its data into that slot. Once that time passes, the node must wait for the net cycle or another frame that is dedicated to them in the same cycle. As shown in the figure 2.7.

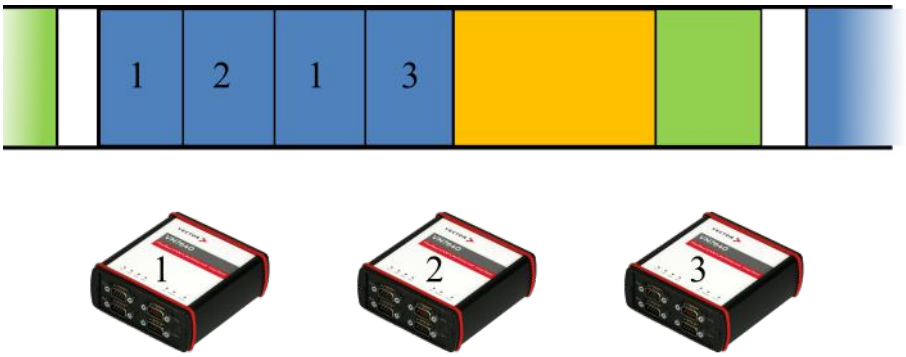


Figure 2.7: Static segment with three nodes.

If a node goes off-line or decides not to transmit any data, its slot remains empty and it is not used by any other node as shown in the figure 2.8.

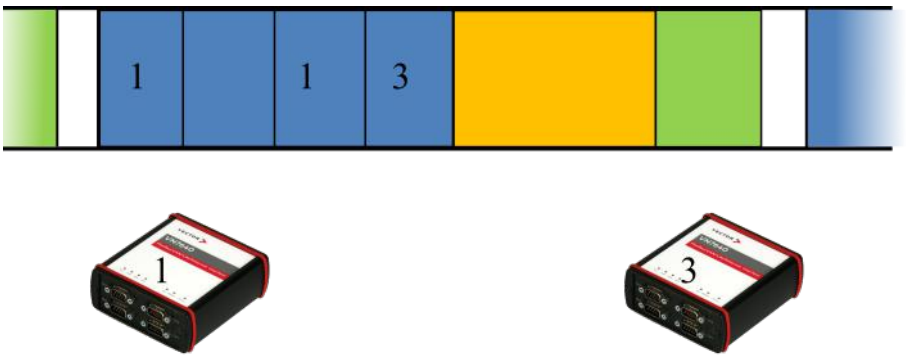


Figure 2.8: Same static cycle with one node off-line.

Dynamic Segment

Most embedded networks have a small number of critical messages and a large number of non-critical messages with a lower sampling frequency. For these low-priority signals it is a dynamic segment, which allows transmitting data occasionally. The segment has a fixed length, so there is a limit of the data that can be transmitted through the dynamic segment per cycle. To prioritize more important signals the dynamic segment is divided into slots. Every slot is pre-assigned to nodes. Higher priority signals receive slots closer to the static segment.

When a slot occurs, an ECU has an opportunity to broadcast data. If it doesn't, it loses its spot and it needs to wait for another slot that is assigned to itself or for the whole cycle to repeat. As we can see in the figure 2.9 there are many unused slots so lower priority ECU can transmit its data (The higher number The higher priority).

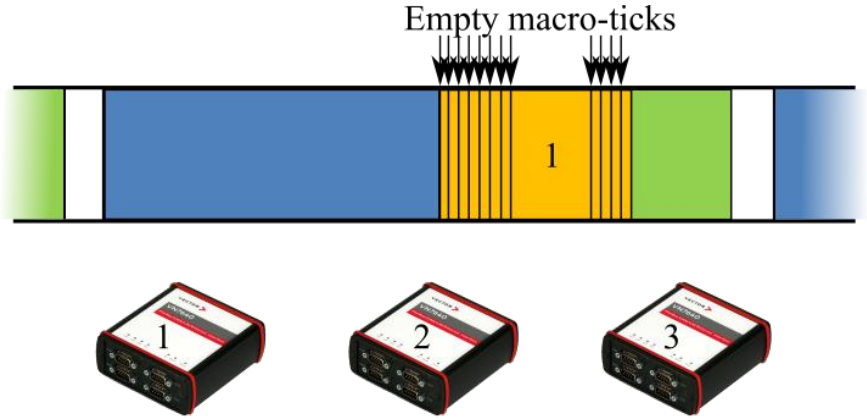


Figure 2.9: Dynamic segment with only one node broadcasting data.

However, when the higher priority CPU transmits its data, it pushes frames of the others. That means there can not be space for lower priority ECU to sending its data. They need to wait for the cycle to repeat and they might get a space to transmit its data as we can see in the figure 2.10. The FlexRay cycle is badly designed when some ECUs cant transmit its data for a more significant number of cycles.

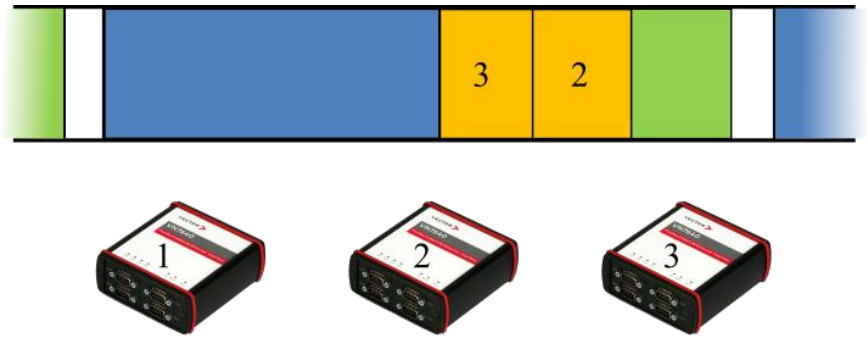


Figure 2.10: Dynamic segment with one node pushed out of dynamic segment.

Symbol Window

The symbol window is primarily used for an identification of special cycles (the cold start - will be discussed later) and for a maintenance. High-level applications do not interact with the symbol window.

Network Idle Time

The network idle time has pre-defined length. The ECUs use this time for calculations.

Data Security and Error Handling

As it was said in the introduction, the FlexRay network can use a single or dual-channel communication. For critical applications devices connected to the network can use both channels. However, when it is need for a more significant bandwidth, nodes can use both buses for transferring non-redundant data. Lastly, if we need to decrease the price of our network and a redundancy is not needed, we can use one channel only. The bus provides an error detection mechanism on the physical layer. This mechanism protects channels from an interference caused by communication that is not aligned with the cluster's communication schedule.

Frame Format

Each slot of the static or dynamic segment contains the FlexRay Frame. As we can see in the figure 2.11.

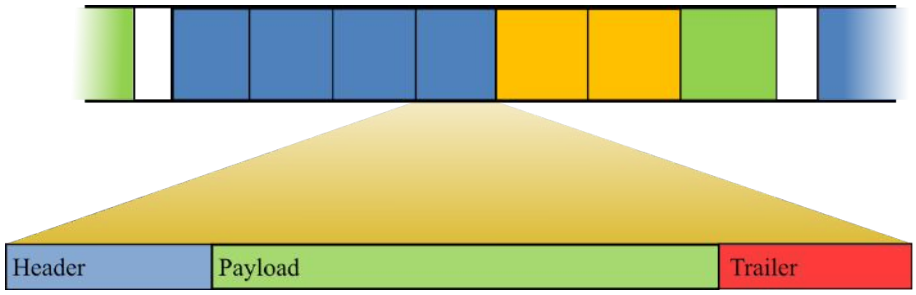


Figure 2.11: Detail of FlexRay Frame.

Header

The Header is 40 bits long and contains of these fields:

1. Status Bits - 5 bits.
2. Frame ID - Defines the slot in which the frame should be transmitted and it is also used for prioritizing in the dynamic cycle, 11 bits long.
3. Payload length - Contains a number of words which are transferred in the frame, 7 bits.
4. Header CRC - Used for detecting errors during the transfer, 11 bits.
5. Cycle Count - Contains the value of a counter that advances incrementally each time when the communication Cycle starts, 6 bits.

All these sections can be seen in the figure 2.12

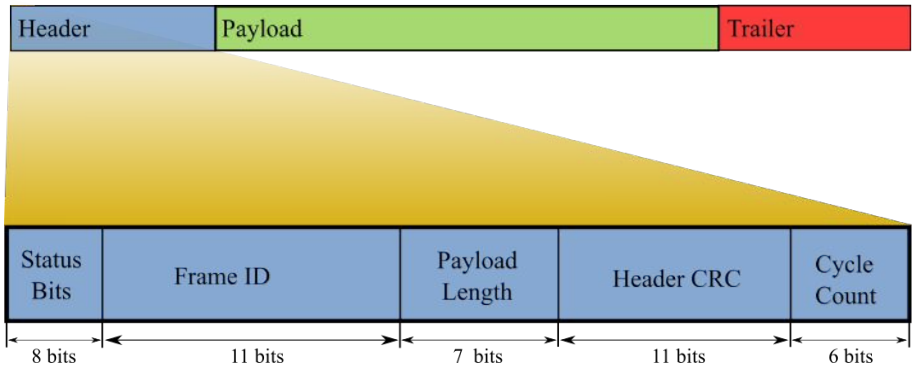


Figure 2.12: Detail of Header section.

Payload

The Payload section contains only data. The length is given header and can vary up to 127 words (254 bytes), which is 30 times higher in comparison with the CAN.

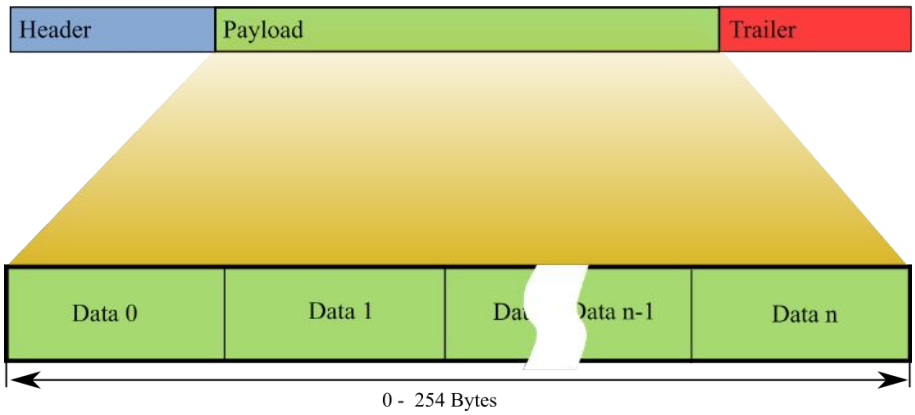


Figure 2.13: Detail of Payload section.

Trailer

Trailer is 24 bits long and contains three CRCs to detect errors.

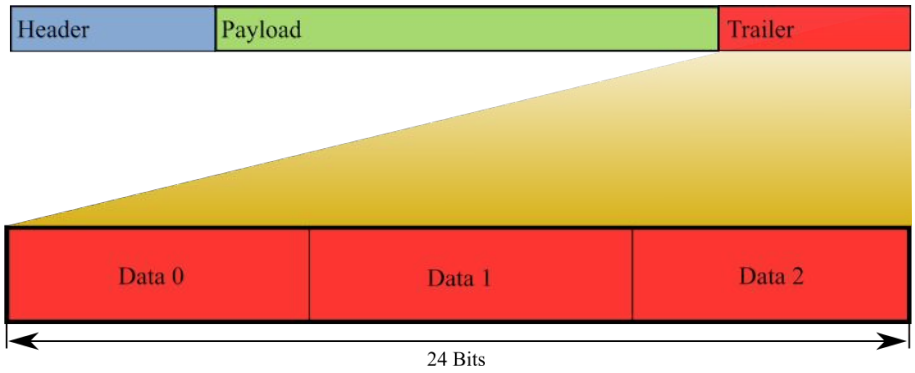


Figure 2.14: Detail of Trailer section.

Signals

FlexRay data is represented in bytes, but most applications require data to be represented in real decimal values with units, scaling and limits. To obtain signals from raw data we need to apply scaling and offset. The scaling and offset values are different for each signal.

To store all these values, FlexRay networks use FIBEX (FIeld Bus EXchange) database that defines the network.

Clock synchronization and cold starting

The FlexRay got the ability to sync up nodes on a network without external synchronization clock signal. This is done by two unique types of frames: Startup frame and Sync frame. To start the FlexRay network, at least two different nodes are needed to send a startup frame. This process is known as the cold-start and the nodes that initiate the start of the network are called the cold-start nodes.

When the network is started, all nodes need to synchronize their oscillators to the network's macro-tick. This is done by two more synchronization nodes. This can be two separate nodes included in the network that are pre-designed to send special sync frames when they are turned on for first time. Other nodes are waiting for these frames and measure the time between these two frames to calibrate their internal clocks to the FlexRay time. After the initial synchronization the network uses the Network Idle Time (Chapter 2.2.2). This part of the cycle has predefined length and every node measures this time to adjust its clock.

FIBEX

The FIeld Bus EXchange (FIBEX) is an XML-based (Extensible Markup Language) standardized file format by the ASAM (Association for Standardization of Automation and Measuring Systems) consortium (citovat). The FIBEX is used for describing automotive networks and is compatible with many different automotive protocols. These databases are produced by vehicle-network designers and are unique for different versions of cars/systems.

The FIBEX can contain many information about a particular network, but it must include the following:

- Transmit and receive schedules.
- Frame definitions.
- Signal definitions.
- Bit-level encoding of signals.
- Network topology.
- ECU information.
- Network configurations including baud rates and timings.

2.3. Introduction to the vehicle modeling

In upcoming section 4.3 of this thesis simulations will be introduced. For these simulations to be correct, we need to develop car models that correspond with reality as closely as possible.

2.3.1. Tire model

The tire modeling and mathematical description of the interaction between the tire and road surface is the biggest challenge. In this thesis, I neglect some tire dynamics, such as the combined slip, influence of the tire deflection, the self-aligning torque. More advanced mathematical models can be found in [9] and [11].

For purposes of this thesis, we use two lines model. That is specified and compared to Pacejka's 'Magic' formula in [6].

Two-lines model

This model represents non-linearity of the tire model by saturation of the linear model to cover a maximal transferable force in each direction. The modified model is given by:

$$f_x(\alpha) = \begin{cases} C_{\alpha 0}\alpha & when\ C_{\alpha 0}\alpha \leq F_z\mu_{max} \\ F_z\mu_{max} & when\ C_{\alpha 0}\alpha > F_z\mu_{max}, \end{cases} \tag{2.1}$$

where $C_{\alpha 0}$ is a side-slip coefficient, α is a side-slip angle, F_z is a normal load on the tire, μ_{max} is a maximal friction coefficient.
Side-slip angle can be defined as follows:

$$\alpha = arctan(\frac{v_y}{v_x}), \tag{2.2}$$

where v_y is a lateral tire velocity, and v_x is a longitudinal velocity. Orientation can be seen in the figure 2.15.

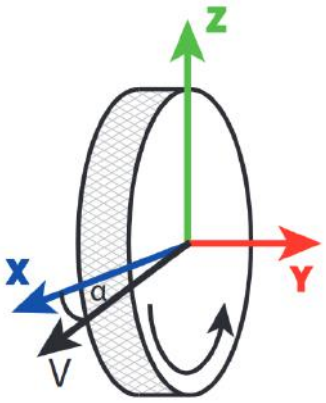
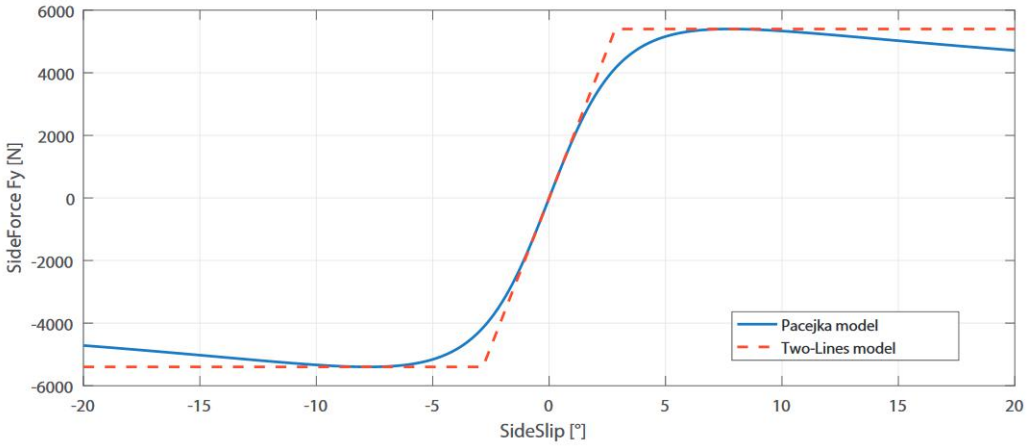


Figure 2.15: Tire coordination system. [6]

Tire model comparison

In [6] we can see a comparison of two tire models in a lateral direction. The first of the two compared models is Pacejka's 'Magic' formula. This tire model is often used for detailed simulations of vehicle dynamics.The second is a linearized model represented by two-lines. As we can see on the figure 2.16 if the side slip angle is $|\alpha| < 2$ we can assume that both models are equal.



Normal force load $F_z = 6000$ N and friction $\mu = 0.9$.

Figure 2.16: Lateral force characteristic for different tire models [6]

2.3.2. Kinematic model

We derive the kinematic model from the figure 2.17. Because our testing vehicle can steer with all four wheels we need to change the basic bicycle model to correspond with reality. In all our equations steering angles are positive as they are shown on the figure 2.17, that is because when we are making a right turn (front wheels directing to right) rear wheels are directing to left.

An intersection of normal to the direction of x-axis of the front tire and normal to the direction of x-axis of the rear tire is point C, which is the center of cornering.The distance between C and the center of gravity is a cornering radius R. From the figure 2.17 we can compute vehicle states which are as follows:

$$\dot{\psi} = \frac{V cos(\beta)}{l_r + l_f}(tg(\delta_f) - tg(\delta_r)), \tag{2.3}$$

$$\beta = tan^{-1}(\frac{l_f tan(\delta_r) + l_r tan(\delta_f)}{l_r + l_f}), \tag{2.4}$$

where v is a vehicle speed, R is a cornering radius, δ_f and δ_r are steer angles of the front and rear tire respectively, l_f and l_r are distances of the centre of gravity of the car to the front and rear axle respectively, $\dot{\psi}$ is the vehicle yaw rate and β is the side-slip angle.

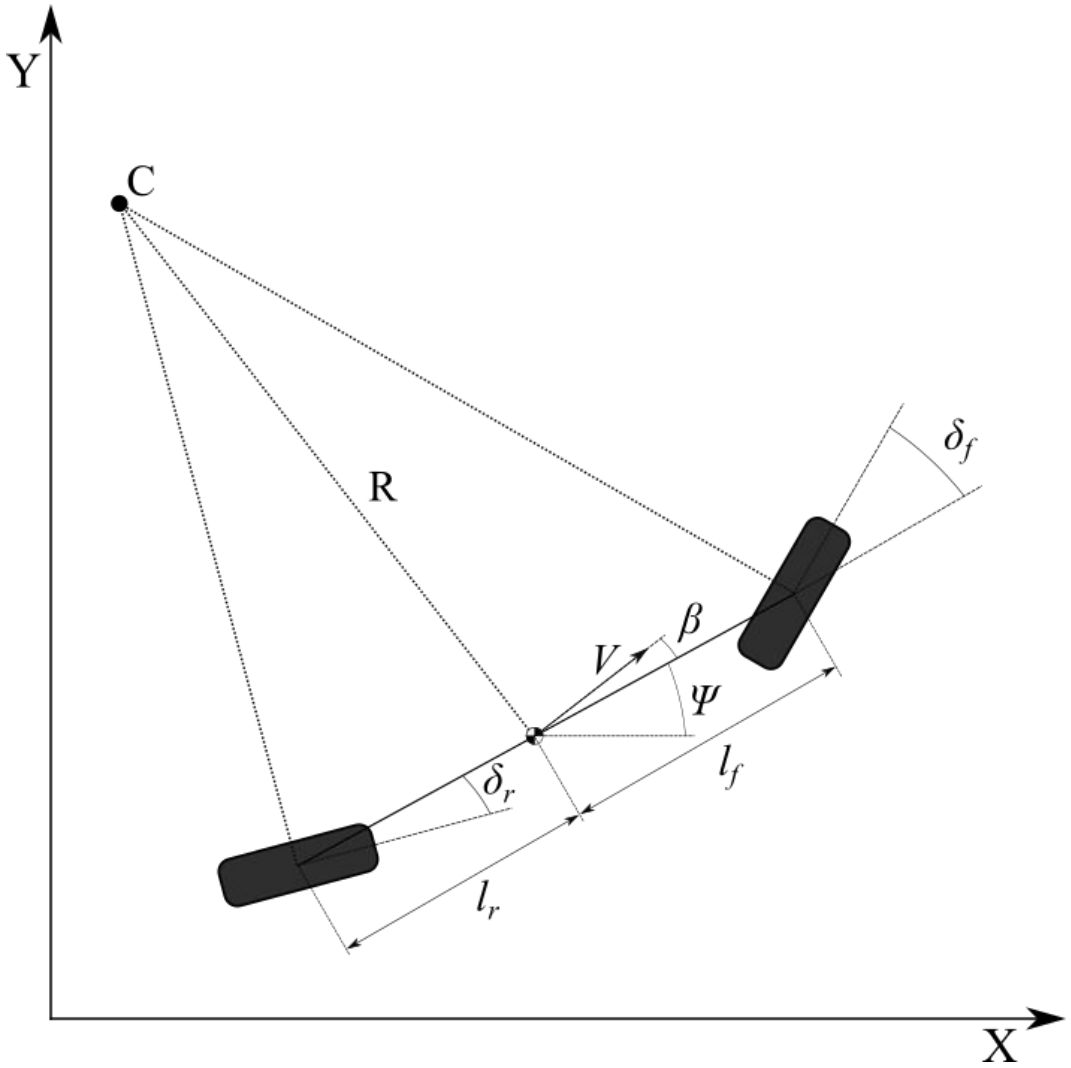


Figure 2.17: Single track kinematic vehicle model

When the steering angles of both wheels are 0, the cornering radius approaches infinity, and the vehicle yaw rate and the slip-angle are reaching 0. More about this vehicle model can be found in [11].

2.3.3. Dynamic single track model

As our test vehicle is a sports car. Therefore CG of the vehicle is near the bottom of the vehicle. Which neglects the load transfer during vehicle motion. Which allows us to use the single track model.

More complex vehicle models are usually used for accurate calculations of a vehicle movement. Using of non-linearized models of parts of vehicle (e.g., tire mechanics, drive train, the position of CG) can be found in [13] [15].

The coordinating system of a vehicle is defined as shown in the figure 2.18, the x-axis is in the direction of forward movement of the vehicle. The y-axis is from CG to the left seen from the driver's point of view. Last is z-axis which is from CG up to fit in a normally used right-handed coordination system.

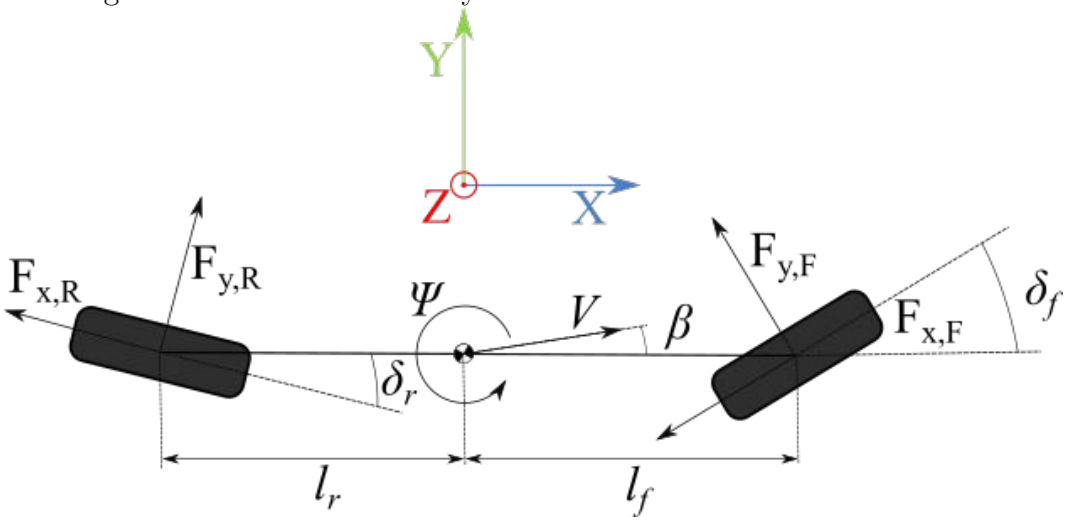


Figure 2.18: Single track dynamic vehicle model

In the figure 2.18 we can see a scheme of single track vehicle model, the main difference is that both front wheels are placed in the middle of center axis and the same principle is used for both rear wheels. The cornering stiffness's of wheels is increased to include the influence of both wheels.

The angle of speed in CG, which is called vehicle side-slip is given by the formula 2.5:

$$\beta = arctan(\frac{v_y}{v_x}), \tag{2.5}$$

where v_y and v_x is the speed in axis respectively to its index.

The differential equations of motion of the vehicle from the figure 2.18 can be derived to 3 axis (translation in x 2.6 and y 2.7, and rotation around z axis 2.8):

$$\begin{aligned} m\dot{v}cos(\beta) + mv(\dot{\beta} + \dot{\psi})sin(\beta) = -F_{x,R}cos(\delta_R) + \\ +F_{y,R}sin(\delta_R) + F_{x,F}cos(\delta_F) - F_{y,F}sin(\delta_F), \end{aligned} \tag{2.6}$$

$$\begin{aligned} m\dot{v}sin(\beta) + mv(\dot{\beta} + \dot{\psi})cos(\beta) = F_{x,R}sin(\delta_R) + \\ +F_{y,R}cos(\delta_R) - F_{x,F}sin(\delta_F) + F_{y,F}cos(\delta_F), \end{aligned} \tag{2.7}$$

$$\begin{aligned} I_z\ddot{\psi} = -F_{x,R}l_Rsin(\delta_R) - F_{y,R}l_Rcos(\delta_R) - \\ -F_{x,F}l_Fsin(\delta_F) + F_{y,F}l_Fcos(\delta_F), \end{aligned} \tag{2.8}$$

where l_f and l_r are the distances of the center of mass from the front or rear axle respectively, δ_f and δ_r are steering angles of the front or rear wheels respectively, m is

the mass of the vehicle, v is the speed of the vehicle, $\dot{\psi}$ is a yaw rate of the vehicle, I_z is a moment of inertia about the z axis, β is the side-slip angle of center of mass, $F_{y,F}$ and $F_{y,R}$ are longitudinal forces of the front or rear tire respectively, $F_{x,F}$ and $F_{x,R}$ are lateral forces of the front or rear tire respectively.

For simplicity we can assume that vehicle will have a constant speed during cornering. Which means that the $\dot{v}$ is equal to zero. Furthermore we will simplify this model more by assuming that steering angles of both(front and rear) wheels are smaller than 10°. Which means that we can linearise the model using equations:

$$sin(x) \approx x, \tag{2.9}$$

$$cos(x) \approx 1. \tag{2.10}$$

Reformulated equations of differential vehicle model are now as follows:

$$mv(\dot{\beta} + \dot{\psi}) - F_{y,R} - F_{y,F} = 0, \tag{2.11}$$

$$I_z\ddot{\psi} + F_{y,R}l_R - F_{y,F}l_F = 0, \tag{2.12}$$

where equation for x-axis is no more taken into account beacause of the assumption of a constant speed. And the eq. 2.11 is in the y-axis and eq. 2.12 is in around the z-axis.

Tire forces $F_{y,F}$ and $F_{y,R}$ are defined by equation 2.1, where α for small steering angles is defined by following equations:

$$\alpha_F = \delta_F - \beta - \frac{l_F\dot{\psi}}{v}, \tag{2.13}$$

$$\alpha_R = \delta_R - \beta + \frac{l_R\dot{\psi}}{v}. \tag{2.14}$$

After substituting of equations 2.13, 2.14 and 2.1 to equations 2.11 and 2.12 we get following:

$$\dot{\beta} = -\beta\frac{C_{\alpha,R} + C_{\alpha,F}}{mv} + \dot{\psi}(-1 + \frac{C_{\alpha,R}l_R - C_{\alpha,F}l_F}{mv^2}) + \delta_F\frac{C_{\alpha,F}}{mv} + \delta_R\frac{C_{\alpha,R}}{mv}, \tag{2.15}$$

$$\ddot{\psi} = \beta\frac{C_{\alpha,R}l_R - C_{\alpha,F}l_F}{I_z} - \dot{\psi}\frac{C_{\alpha,F}l_F^2 + C_{\alpha,R}l_R^2}{vI_z} + \delta_F\frac{C_{\alpha,F}l_F}{I_z} - \delta_R\frac{C_{\alpha,R}l_R}{I_z}. \tag{2.16}$$

After small manipulation we can modify this linearized differential equations into following state space model:

$$\begin{bmatrix} \dot{\beta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} -\frac{C_{\alpha,R}+C_{\alpha,F}}{mv} & -1+\frac{C_{\alpha,R}l_R-C_{\alpha,F}l_F}{mv^2} \\ \frac{C_{\alpha,R}l_R-C_{\alpha,F}l_F}{I_z} & -\frac{C_{\alpha,F}l_F^2+C_{\alpha,R}l_R^2}{vI_z} \end{bmatrix} \begin{bmatrix} \beta \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha,F}}{mv} & \frac{C_{\alpha,R}}{mv} \\ \frac{C_{\alpha,F}l_F}{I_z} & -\frac{C_{\alpha,R}l_R}{I_z} \end{bmatrix} \begin{bmatrix} \delta_F \\ \delta_R \end{bmatrix} \tag{2.17}$$

For simulation purposes, verifying and visualization purposes we introduce the following equations, that will provide us with projection to global X and Y which we can see in figure 2.17.

$$\dot{x} = V cos(\psi + \beta) \tag{2.18}$$

$$\dot{y} = V cos(\psi + \beta) \tag{2.19}$$

From equation 2.18 we get speed in the X-axis, to get X position we need to integrate it. The same is done for equation 2.19. Both of these equations work for dynamic model ass well as for kinematic.

3. Goals and design of thesis

Before any practical work on this thesis is started, analysis and planning needs to be done. This chapter is focused on steps that are required to be taken to finish this thesis.

Firstly, measurements are needed to determine which signals are responsible for the moment of the ESP and thus which signals are responsible for steering. However, signals about position, speed of the steering wheel, and torque on the steering wheel cannot be used because they are processed in the LKA control unit, and thus cannot be modified before they take effect on front wheels. Because of this reasons series of test will be made on the highway with the lane keeping system turned on to determine which signals are responsible for an automatic front wheel steer. This will be done by logging every signal in the vehicle and comparing them to events that happened with the vehicle on the road.

The next step is to make sure that the test vehicle is able to drive normally when the designed system is turned off. This means that when the autonomous steering is turned off the vehicle has to behave normally, so the driver feels no difference. This has to be done as a first modification of the vehicle, because there are powertrain components on the same branch of the vehicle and thus are critical for correct function. Also, safety features will be added to make sure everything is as safe as possible, in our case whenever the driver touches the brake pedal every component of the vehicle will start working as it was a mass-produced car.

The next step is the design of the controller on its own. To drive the test vehicle through the test track, we need several layers of control. We start with the vehicle lowest level of control which in our case is the steering angle of wheels. Because of the complex behavior of the front wheel axle and front wheels, feedback PID regulator was chosen. The situation in this state can be seen on the diagram in the figure 3.1, where the elements of the vehicle are shown with the feedback and PID regulator.

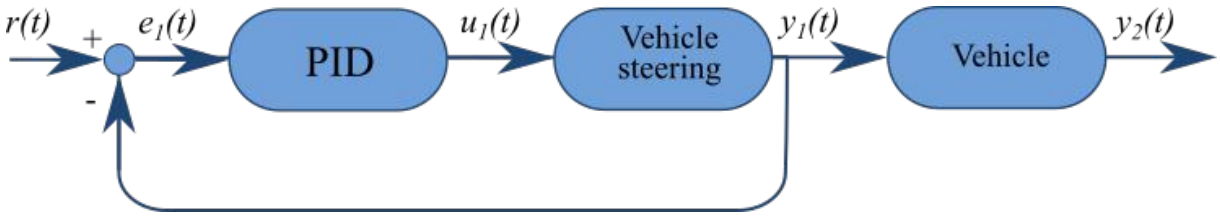


Figure 3.1: Scheme of front wheels steering angle controller with PID regulator.

Where the r is desired front wheel angle, e_1 is difference from actual and desired front wheel angle, u_1 is a control variable, which is moment, y_1 is an actual front wheel steering angle and y_2 composite from states of the vehicle($\dot{\psi}$, $\ddot{\psi}$) and position(x, y).

The next layer is a control of the position of the vehicle. This will be done in several ways. The first is the PID regulator with the Y position error input, given that we follow the coordinate system introduced in the figure 2.18. To avoid problems with the inability to reduce error to zero we can add a position shift of the trajectory and then the scheme would look like it is shown in the figure 3.2.

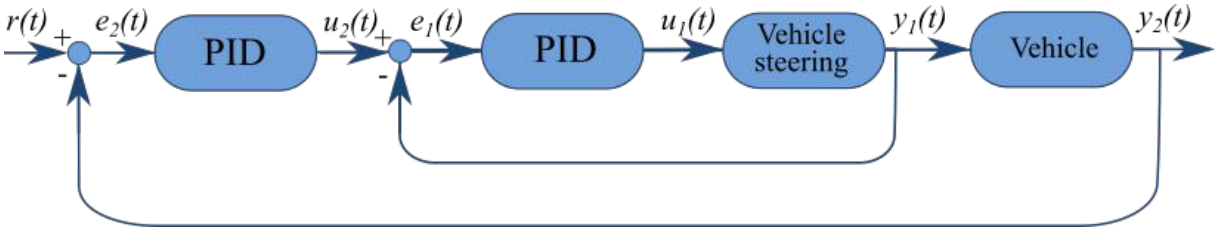


Figure 3.2: Scheme of position controller with only PID regulator.

Where the r, in this case, is the desired position along the Y and X axis, e_2 is the difference from actual and desired position along the Y-axis, u_2 is a control variable, which is in our case the steering angle. The rest is the same as in 3.1.

The output of the vehicle is measured by vehicle sensors in the case of $\dot{\psi}$ and $\ddot{\psi}$, and by differential GPS in case of position.

Because of the lower frequency of measurements of the GPS, the next step is to make a mathematical model that will be used as a prediction to the Kalman filter, which will be implemented later. This means that the mathematical model from the section 2.3 needs to be parametrized and verified. The control scheme with the Kalman filter and mathematical model of the test vehicle is shown in the figure 3.3

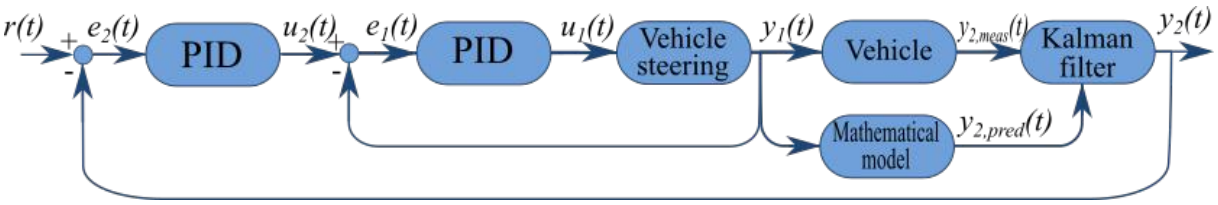


Figure 3.3: Scheme of position controller with with kalman filter.

Where y_2 is calculated the output from Kalman filter, $y_{2,pred}$ is the predicted output from the mathematical model and $y_{2,meas}$ is the measured output of the real vehicle.

The last step, to ease the work of PID is to add feedforward calculation of the steering angle of the front wheels. This will be done with the help of equations from the section 2.3. In the ideal case, this would remove the need of PID feedback regulation, but our model is not perfect, and we do not take in factor disturbances from the tarmac, and other factors, which means that there will be always some action needed from PID regulator. Scheme of the whole controller is shown in the figure 3.4

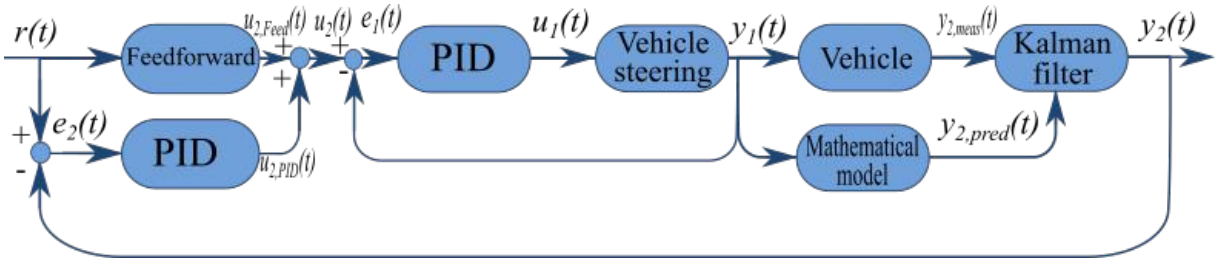


Figure 3.4: Scheme of position controller with with kalman filter.

Where the $u_{2,PID}$ is the control variable of PID regulator, $u_{2,Feed}$ is the control variable of Feedforward algorithm, u_2 is sum of PID and feedforward output and other are same as before.

Because of the limited time that can be spent with the test vehicle on test polygon everything has to be tested before it is tested on the real vehicle. This will be done for all three types of control (Feedforward, PID, Feedforward + PID), and for situations where limitations of the actual vehicle will not be taken into account. Also, simulations with restrictions will be done to determine the maximal speed of the vehicle on which it can drive through the maneuver.

The last steps are the testing and evaluating the results. The testing will be done on the test polygon with the real test vehicle, with the help of differential GPS lent by CIIRC VUT. In the Feedforward test, GPS data will be recorded only for analysis of results, but for PID and Feedforward + PID test, GPS data will be used to measure position for Kalman filter. The results will be compared to simulations and to each other to determine the best form of control.

4. Design and implementation

This chapter is dedicated to description of practical steps, that are needed for successful finish of this thesis.

4.1. Design of signal gateway

For our purposes, we need to modify car steering. This will be done by cutting the communication network within a vehicle and place a unit that can edit messages within FlexRay signals.

First, we need to know which ECUs are responsible for calculating necessary steering angles, than which signals are used to transfer this information, also which ECU is receiving these signals. We must not forget about the properties of these signals and inputs of ECU.

In this section, I will try to find answers, to all essential questions about these signals, and their limitations.

4.1.1. Lane keeping assistant

For purposes of steering with wheels, we will use a system called lane-keeping assistant, which is implemented in our car. This system is made as a safety feature as it will steer the vehicle back to the lane when it drifts out of lane [7].

This system is located in the HCA control unit, containing cameras that capture lanes on the road. This data is then used to calculate if any corrections are needed and how much we need to correct the angle of the vehicle. These values are then sent to the part of the vehicle which can make necessary changes. For our steering controller, it's a unit that controls electric power steering and its called EPS. This ECU does not control electric motor only based on inputs from the steering wheel but also outputs by other ECUs, which in our case is the HCA control unit which takes responsibility for lane keeping assist.

Vehicle network

In our testing vehicle, we have a hybrid network topology 2.2.2, which is shown in figure 4.1 with our HCA and EPS unit marked. This topology gives us few options where to modify signals from HCA, and first place is on the left side of the HCA control unit just before signals enter the central core of the network. Or on the left of EPS ECU.

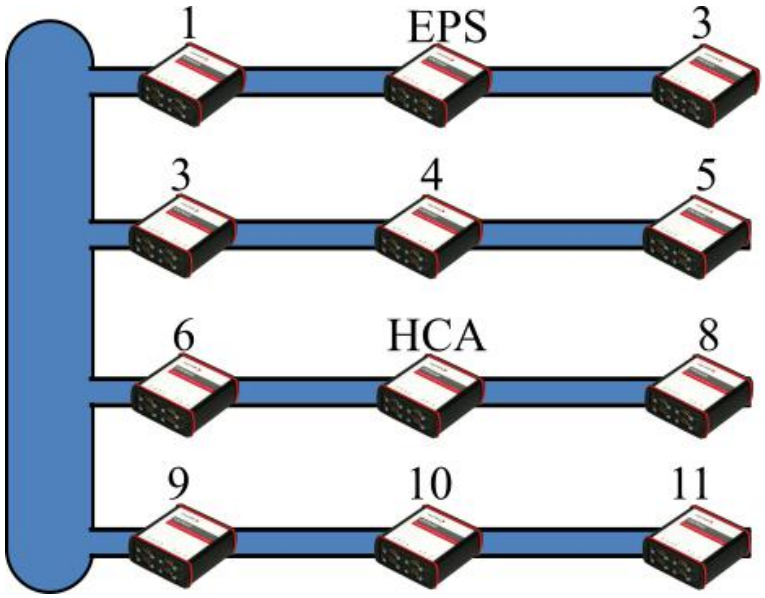


Figure 4.1: Testing vehicle FlexRay network.

The first place is not very suitable because frames which contains our needed messages are read by other control units as well, and we can not be sure if some of our modifications of messages would not unsettle other vehicle units. From this reason, second variant, position in between ECU 1 and EPS ECU, was chosen.

After our modifications car network looks as it is displayed on the figure 4.2

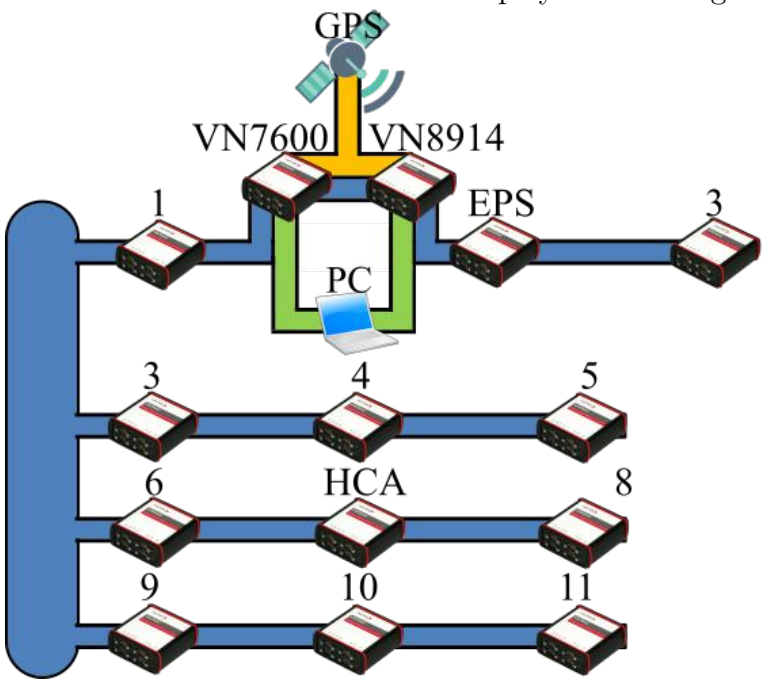


Figure 4.2: Testing vehicle FlexRay modified network with added CAN network and USB communication with PC.

We added two more ECUs to the FlexRay network. First is VN7600 which is responsible for editing messages we want to edit and work as a gateway for everything else. VN7600 is also connected to PC via USB cable, and to differential GPS via CAN network. The second added ECU on FlexRay network is VN8914 which is mainly used for reading

data from every network that it is connected to. Next, we added two more networks. First is CAN network which connects differential GPS, VN7600, and VN8914, second is USB network which connects PC, VN7600 and VN8914. In figure 4.3 can be seen boot of the test vehicle, with vector VN7600 and VN8914 installed. Power supply was also installed to power all necessary hardware. Personal computer is connected in the front seat of the vehicle.

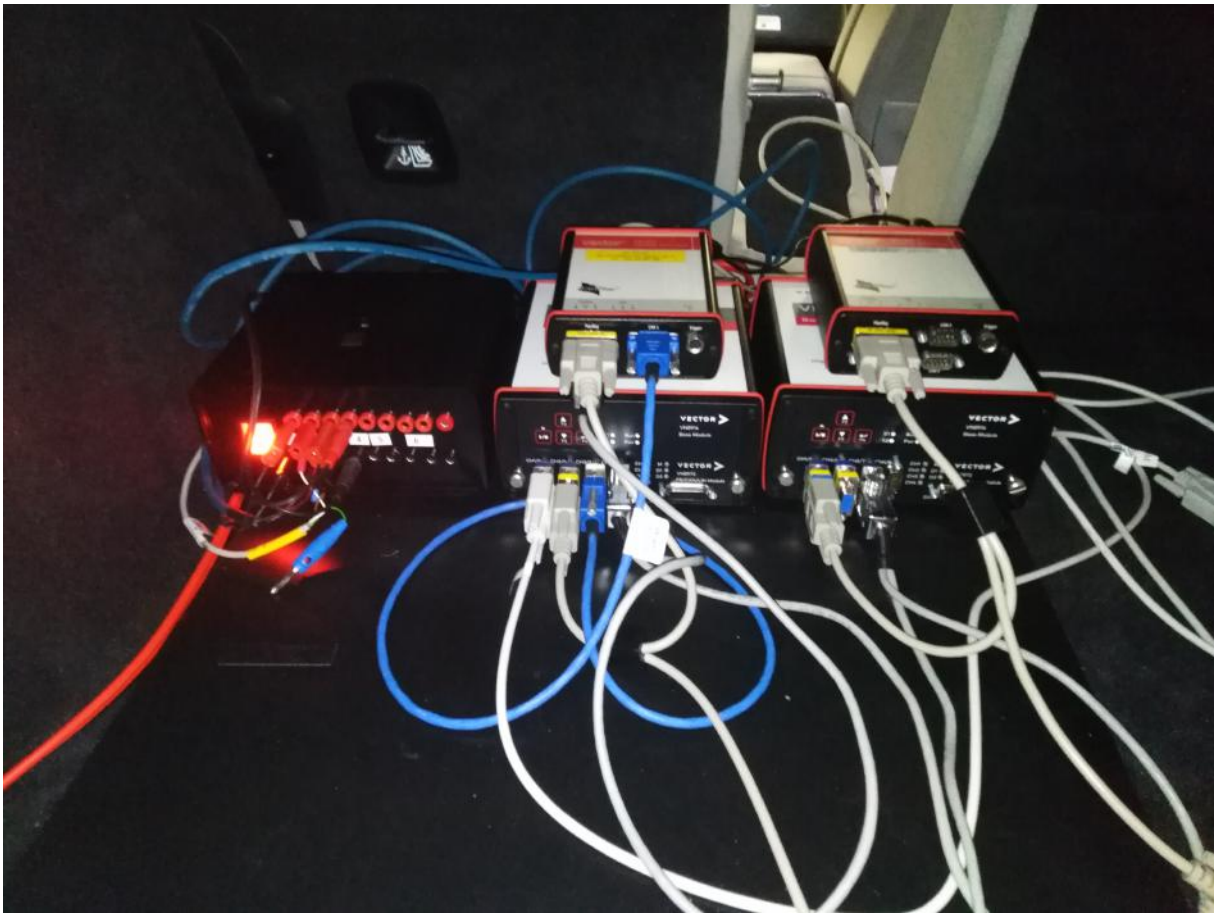


Figure 4.3: Photo of the boot of the testing vehicle.

This new network design was chosen because in future there will be added the same control units for longitudinal dynamics, which means that the same topology will be made on another branch. And for communication between VN boxes will be used CAN network.

VN7600

VN7600 is mainly used for editing messages on FlexRay, but because on the same branch are critically important ECUs for proper vehicle function, VN7600 also needs to work as a gateway for signals that are read by ECUs after modifications are done. Car is controlled from PC that is connected to VN7600 by USB, which means that the VN box is also reading commands from PC and deciding when and what to modify based on these commands. This functionality is done within a Vector interface called CANoe where is used a programming language called CAPL in figure 4.4 we can see an example of gateway code for frame X Y Z, where X Y Z is the address of a frame. After we find which

messages we need to modify, we implement modification code for a frame that contains our messages.

```
5  on frFrame msgchannel1.FRAME_x_y_z
6  {
7    frFrame msgchannel2.FRAME_x_y_z frame_x_y_z_sec;
8    int i;
9    for(i=0;i<FRAME_LENGHT;i++)
10   {
11     frame_x_y_z_sec.byte(i) = this.byte(i);
12   }
13   frOutputDynFrame(frame_x_y_z_sec);
14 }
```

Figure 4.4: Example gateway code for frame x y z.

Also, checks for limitations that were found in section 4.1.2 are implemented here because only this system works perfectly synchronized with the FlexRay network.

VN8914

VN8914 is mainly used for logging data from FlexRay and CAN network. This VN box uses a USB cable to send data to PC where they are stored for later evaluation of the test, or they can be used right away for calculations as we will have our controller running on PC. This unit is also used to check if everything is running correctly as it is checking for error signals from other ECUs on the FlexRay network. Before modification was introduced in our system, this unit was used to obtain signals which are needed to be modified.

4.1.2. Modified signals

We find out that three messages, which are involved in changing moment on steering wheel (therefore moment on front wheels), that are sent from HCA unit to EPS. First is status, this message sends the state of LKA, default value is 0, which means that LKA is waiting to be activated by the driver. The error value is -1, which means that LKA needs to be turned off and that will reset its value to 0, and the last value is 1, which means that LKA wants to be active. The second signal is a moment value that is requested by LKA, and it has values from 0 to 12 newtons. Last message is sign of moment value, which means that if LKA request - 5 Nm, first message is equal to 1 (active LKA), the second message is equal to 5 and the third message is equal to 1, as value 1 means that moment is negative and value 0 means that moment is positive.

From the EPS unit to the HCA unit are send two critical messages. First is again status value, where 0 means that the unit is not listening to the requested moment, one implies that LKA is controlling moment on steering wheel, and -1 is again error value. The second message has information about how long the driver is not holding the steering wheel. This is implemented for safety reasons and after value reaches predefined value, HCA status message will switch to 0.

Signal limitations

After testing, we found out limitations which when not complied, will turn off LKA or push it to error. The first limitation is that we need to wait until we get a response from the EPS unit about listening to our desired moment before we send a value of the moment. This usually happens in the next cycle. We implemented a function in VN7600 that will stop sending values of the moment until the correct answer is received.

The second limitation is for the maximal value of a moment that can occur on the steering wheel. If we cross this value EPS unit will set a status message to error value and stops applying moment on the steering wheel. This value is 3.2 Nm, and it is not critical for our purposes as we are able to turn the steering wheel all the way to sides. To avoid malfunction we implemented saturation in VN7600.

The third limitation is for minimal speed that vehicle needs to drive before any modifications can occur. Until we reach this value any modifications will set error values form HCA and EPS to -1 and before any other modifications our controller need to be restarted. Idea of modification speed signal so that vehicle thinks its driving faster than it in reality do was denied. This was because of transmission and other power train control units that are in the same branch as our modifications. Limit speed is 5.1 km/h.

The last limitation is for us most critical, when the derivation of the moment which is sent to EPS is higher than a certain value control unit will set a status message to error value and stops applying moment on the steering wheel. Maximal derivation of the moment is 0.1 Nm, and this is critical for our purposes as it will slow down our maneuverability at higher speeds. However, we implemented a rate limiter in the VN7600 to avoid errors and loose of control.

All limits were tested for different speeds as the speed of the vehicle can have influence on the limits, but our finding is, that all limits are not dependent on the speed of the vehicle.

```
4    // check for minimal speed
5    if (speed > 5.1)
6    {
7      // data loading from Simulink
8      moment_ref = @sysvar::ADP::Offset;
9      signum_ref = @sysvar::ADP::OffSign;
10     // rate limiter
11     if (abs(moment_wanted-moment_ref)>=incrementLMMAX)
12     {
22     else
23     {
26     //
27     moment_action = abs(moment_wanted);
28     moment_action_sign = 0;
29     if(moment_wanted < 0) moment_action_sign = 1;
30
31     // check for maximal value of moment
32     if (moment_action > maxmoment)
33     {
36     //  manipulation HCA_01
37     HCA_sec.HCA_CRC                    =  this.HCA_CRC;
38     // modification of signals in frame
39     HCA_sec.HCA_Status_HCA             =  LKA_RUNNING;
40     HCA_sec.HCA_LM_Offset              =  moment_action/this.HCA_01_LM_Offset.factor;
41     HCA_sec.HCA_LM_OffSign             =  moment_action_sign;
42     // gateway for non modified signal in same frame
43     HCA_sec.anothersignal              = this.anothersignal;
44     //recalculation of CRC for new frame values
45     scode = scodes1[HCA_sec.HCA_01_BZ];
46     HCA_sec.CRC = createCRC(HCA_sec.fr_payload, HCA_sec.fr_PDULength, scode);
47   }
48   // output to FleXRay
49   frOutputDynFrame(frmXYZ_sec);
50 }
```

Figure 4.5: Example of modification of signals and check for limitation.

4.2. Design of PID regulator

As the first layer of regulation, there is a regulation of steering wheel angle by a moment. This can be seen in the figure 4.6 where r is a desired front wheel angle, e_1 is a difference of actual front wheel angle and desired one, u_1 is a control variable, y_1 is an actual front wheel steering angle, y_2 are states of the vehicle.

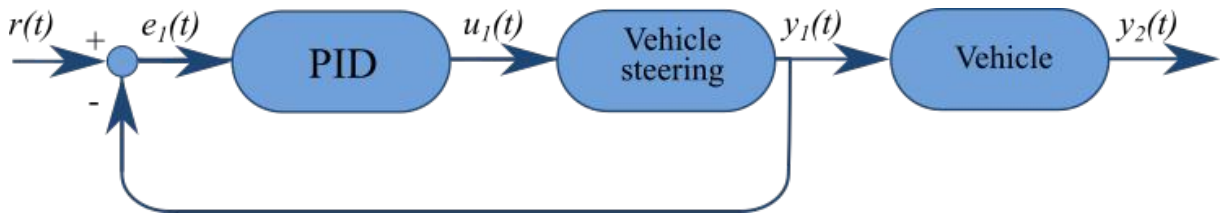


Figure 4.6: System regulator scheme

In this part, I focus on setting the PID regulator and measurements that we need for making a proper design of the PID regulator.

4.2.1. PID regulator scheme

For our purposes we tried two variants of PID regulator, these variants differ only in the way how they solve the problem with integration windup. The first is the back-calculation anti-windup. The scheme for this PI version of this regulator can be seen in the figure 4.7.

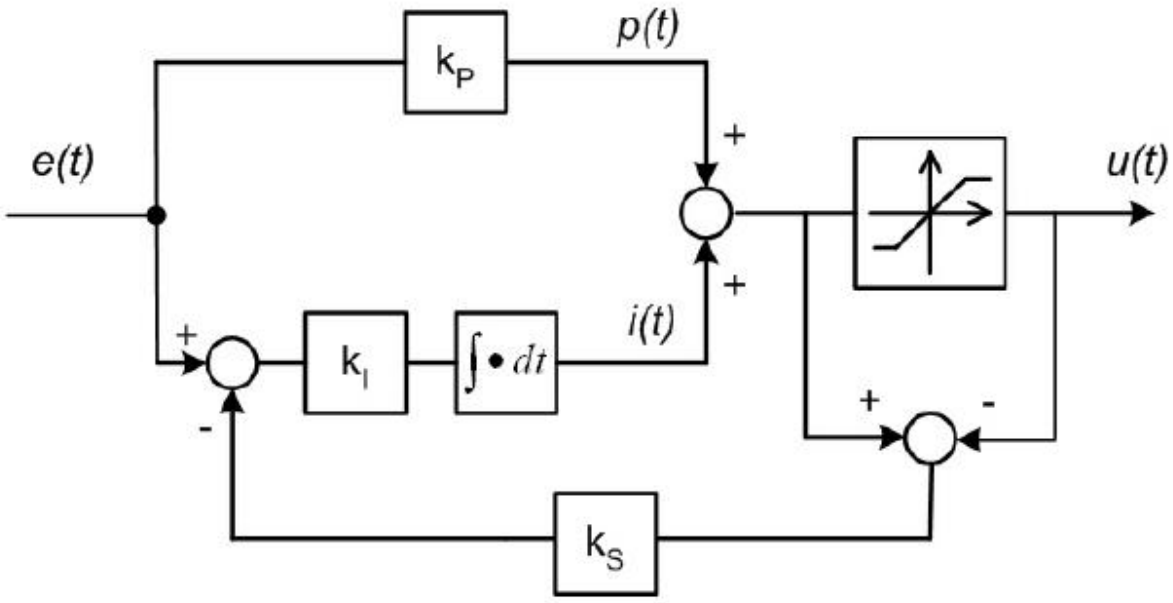


Figure 4.7: Tracking anti-windup, back-calucalation [1]

The second variant, clamping anti-windup can be seen in the figure 4.8.

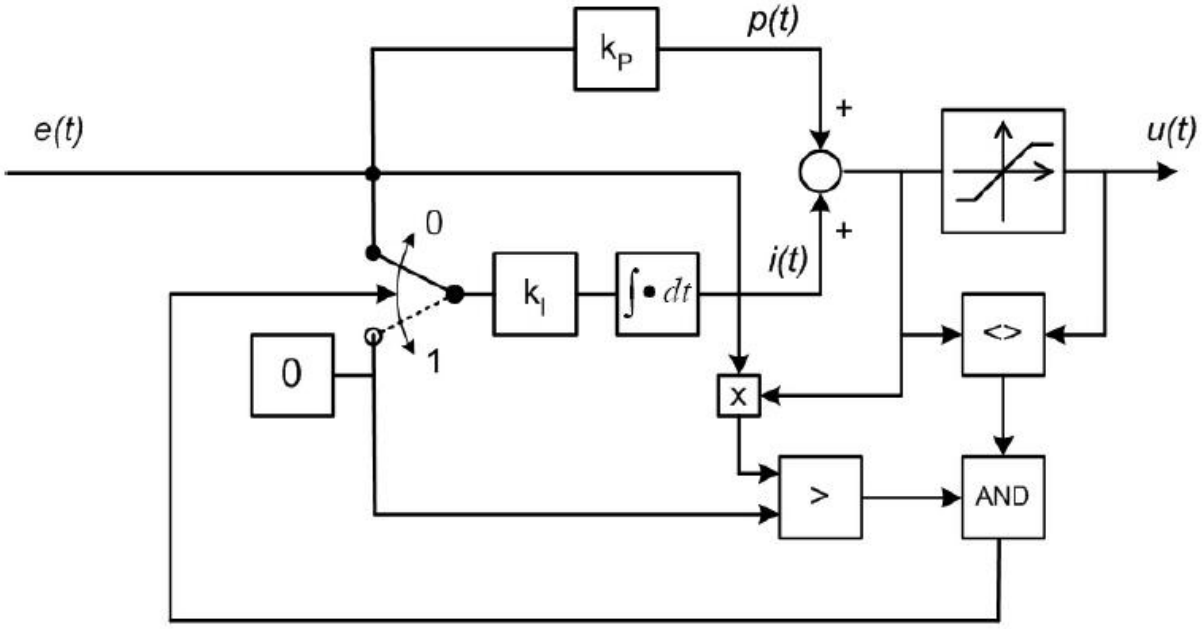


Figure 4.8: Tracking anti-windup, clamping [1]

Testing

To choose which one of these two methods is better we have to do several tests to determine the quality of regulators. The first test is the reaction of the system when the step signal is put as a reference for the steering wheel angle. This can be seen in the figure 4.9

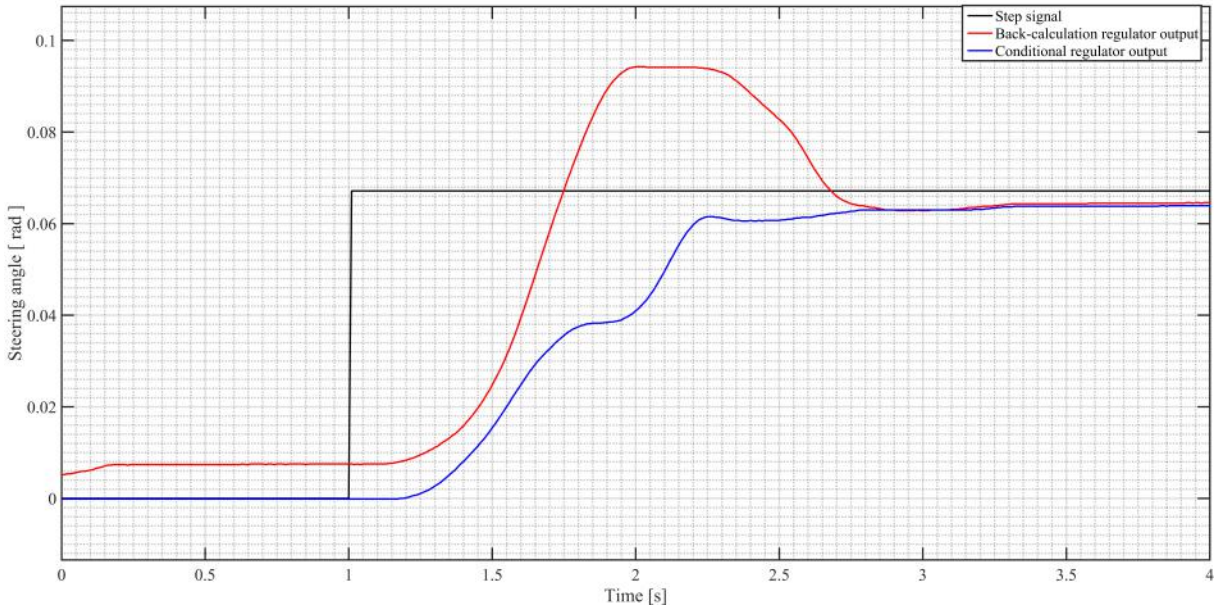


Figure 4.9: Comparsion of back-calculation and clamping PID regulator

As we can see even when the derivation part was set to zero we have a problem with an oscillation in the clamping regulator. On the other side, we have the back-calculation, which is smoother and faster. This test is one-sided enough for us that in following parts we will use only back-calculation scheme, final Simulink version can be seen in the figure 4.10

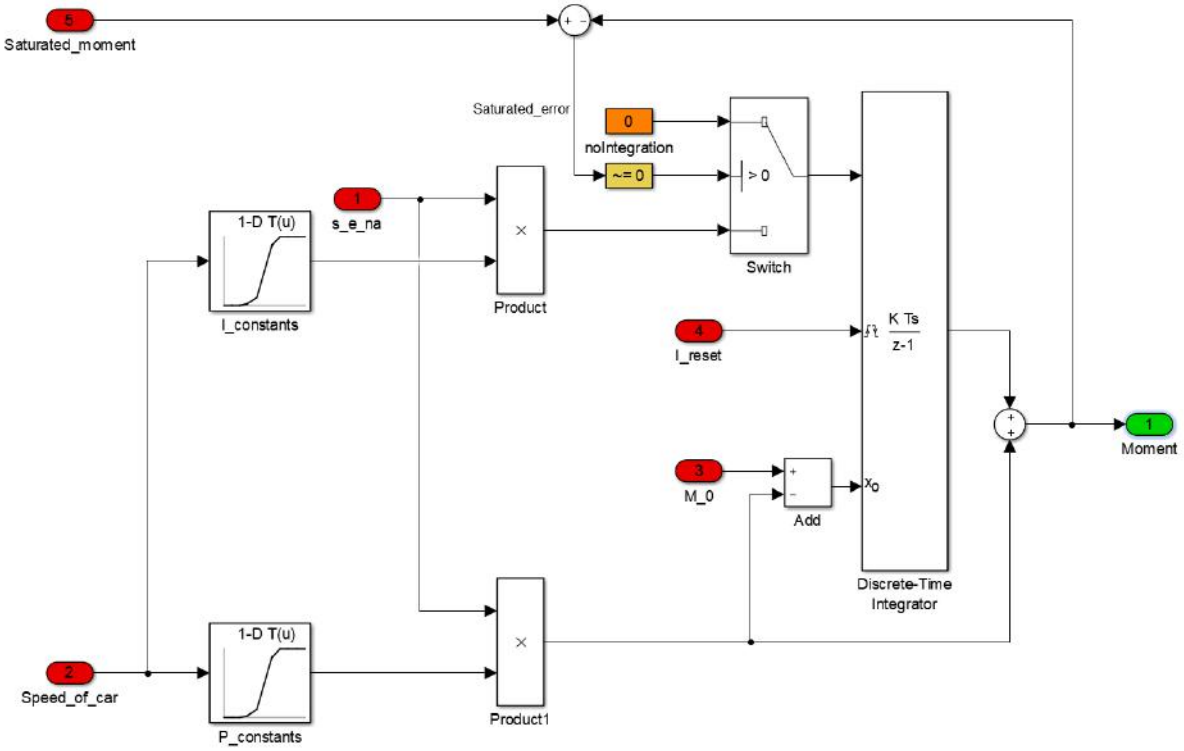


Figure 4.10: Simulink scheme of PI regulator with changeable P and I values.

4.2.2. Setting constants

To proper functionality of the PID regulator, we need to set P, I and D constants. This will be done by the testing in the vehicle by the trial and error method. The principle of this method and steps can be found in [10]. In the figure 4.11 we can see some of our results for the speed 4.1 ms^{-1} we did same tests for different speeds because as the speed of car changes, parameters of steering change as well. Constants will be saved to the look-up table as we can see in the figure 4.10.

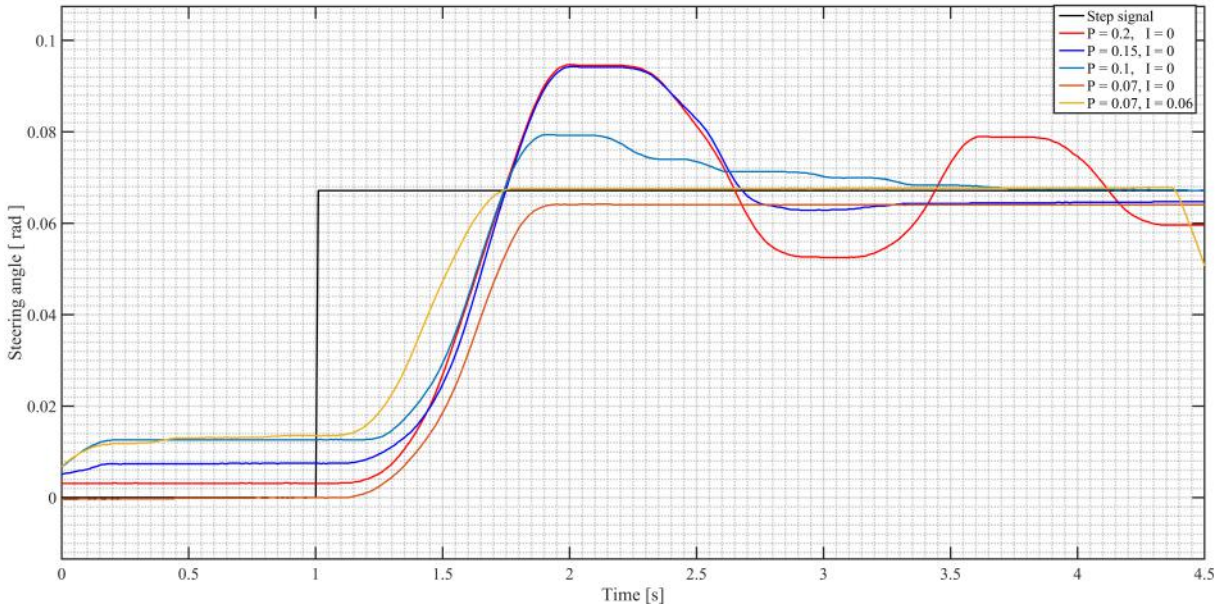


Figure 4.11: Step responses for different P and I values.

4.3. the vehicle parametrization

The primary goal of this thesis is the implementation of the control system on an actual test car. The single track vehicle model introduced in 2.3 must match the real dynamic properties of the test vehicle not only to obtain good simulation results but also to support better design of the control system.

In this section, the test vehicle is introduced, and the parameters of the test vehicle are estimated or measured. Then all parameters are validated with the simulations and vehicle tests.

4.3.1. Test vehicle

The test vehicle used for validating our models, control algorithm and measuring is Porsche Panamera 4.8 Turbo and it can be seen in the figure 4.12. It is a serial car where the only modification is that all FlexRay branches are cut for our purpose of modification of signals. All parameters in the following sections are measured or tuned to match the behavior of our test vehicle.

Figure 4.12: Testing vehicle on testing polygon.

Known vehicle parameters

In this part, I focus on the parameters of the vehicle that are commonly known or can be found in the basic vehicle description. The summary of these parameters can be found in the table 4.1

Total vehicle weight m	2320 kg
Wheelbase l	2950 mm
Moment of inertia along the z-axis I	3600 kgm^2

Table 4.1: the test vehicle parameters [3]

Where total weight is the sum weight of the vehicle, and 250 kg is the weight of three passengers and equipment, which is in the vehicle when the tests are done.

Experimentally identified parameters

Experiments are needed for an approximation of other parameters in the vehicle. This section is focused on experiments and the results of these experiments, which help us to estimate the parameters of the vehicle.

Front steering coefficient

The test vehicle can measure the only angle of the steering wheel, but for our purposes we need an angle of the front wheels. This conversion is non-linear due to the design of the front wheel axle. A test at a low speed was suggested to obtain this dependency. In this test, we drive as slow as possible in circles and measure positions along the way. This measurement can be seen in the figure 4.13. There is snow chain mode turned on in the vehicle, which disables rear axle steering, which means that our testing vehicle now behave like the regular front steered vehicle.

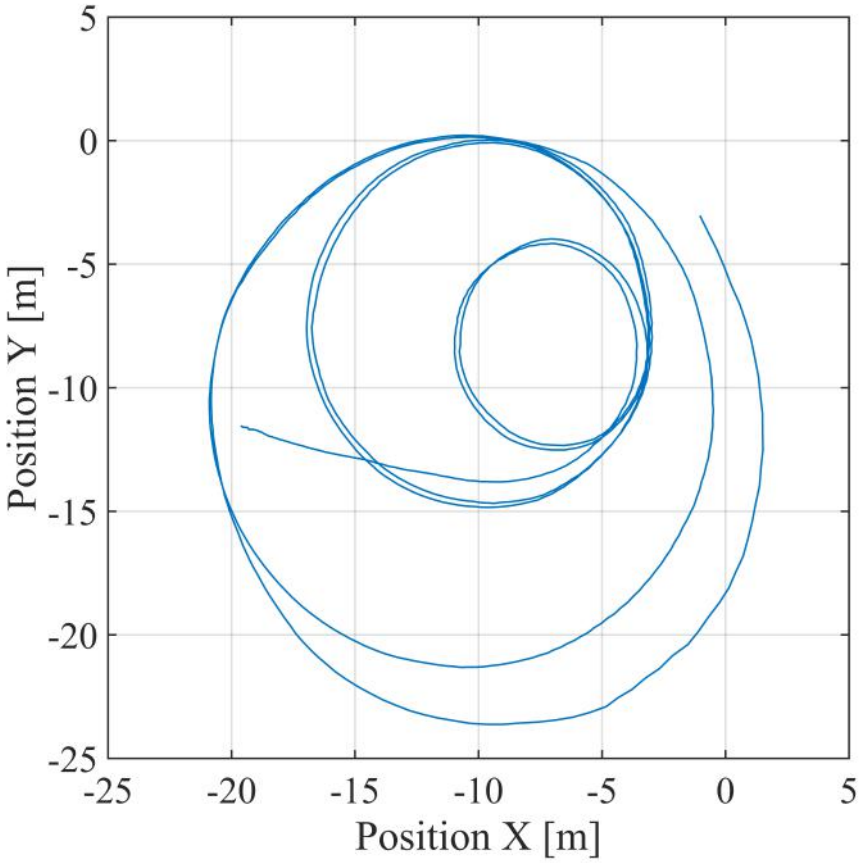


Figure 4.13: Position of the test vehicle during the test with the snow chain mode on.

From the graph there was the radius of each circle measured to determine the turning radius of the test vehicle. And the steering angle of front wheels is calculated from kinematic equations 4.1.

$$\delta_f = arcsin(\frac{l}{R}), \tag{4.1}$$

Where δ_f is the front steering angle, l is the length of the testing vehicle and R is a turning radius of the vehicle. From the steering wheel angle (δ_w)and front wheel angle (δ_f) we can easily calculate the coefficient (k) 4.2

$$k = \frac{\delta_w}{\delta_f} \tag{4.2}$$

After more measurements, we get whole non-linear characteristic of conversion between the steering wheel and the front wheel angle, which looks like in the figure 4.14.

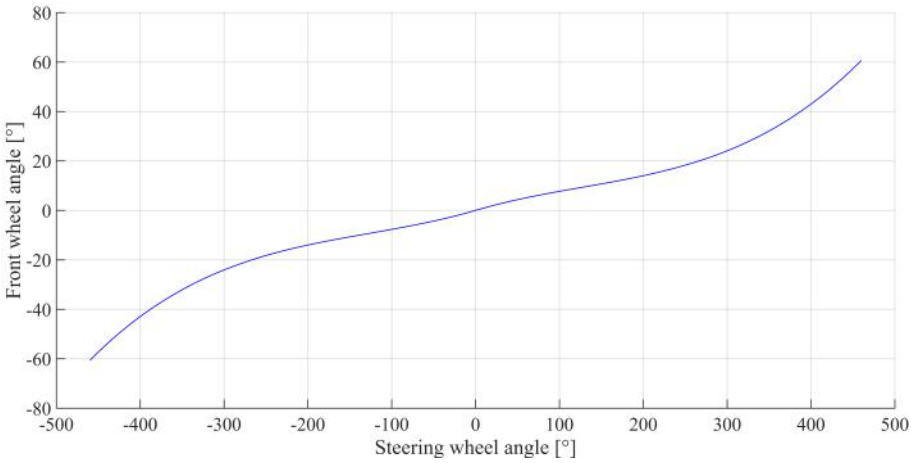


Figure 4.14: Non-linear characteristic between steering wheel angle and front tire angle.

Center of gravity

The next logical step would be to find the steering angle of the rear steering wheel, but for that it is needed to know the position of the center of gravity. Because our testing vehicle is a sports car, we can assume that CG is in the middle, but we can easily measure it. This test is in slow speed again which means that there is no slip on the front and rear wheels. Also because we do not know the steering angle of rear wheels we will turn on the snow chain mode, which disables rear wheel steering. For this test we need to see the vehicle yaw rate $\dot{\psi}$, vehicle speed v and the front wheel steering angle δ_f which will be calculated from the steering wheel angle and conversion. All data will be measured by car control units and will be recorded by the VN8914 box.

From equation 2.3 we can express form 4.3, which will give value of β

$$\beta = acos(\frac{\dot{\psi}L}{vtan(\delta_f)}). \tag{4.3}$$

This can be used to calculate the distance of CG from rear axle using the equation 4.4, which was expressed from 2.4

$$l_r = \frac{Ltan(\beta)}{tan(\delta_f)}. \tag{4.4}$$

This test was done for more steering angles, and for speeds lower than $1.4ms^{-1}$, but every time with a constant speed and a constant value of $\dot{\psi}$. Our tests shows that prediction was correct as we get value of $l_r = 1459mm$, which is 49.42% of total distance L.

Rear steering coefficient

The rear wheel angle depends on more things than only on the position of the steering wheel, the most common thing that changes the behavior of the rear axle is the speed. When the velocity exceeds $60kmh^{-1}$ rear axle starts steering in the opposite way than to this limit speed. This is done to prevent oversteer in the vehicle. Oversteer in the vehicle is when the rear wheels cannot follow circulate trajectory any longer. If oversteer happens in a lower speed rear axle changes again the sign of rear steering angle δ_r.

To find the angle of the rear steering wheel, we drive at a low speed to satisfy the condition of the kinematic model, which means there is no changing of the sign in the rear steering angle.the snow chain mode is turned off to allow rear wheels to turn and series of test are made to measure the rear steering angle. As it can be seen in the figure 4.15 different diameters of the circles are made to calculate rear steering angle for different steering wheel angle.

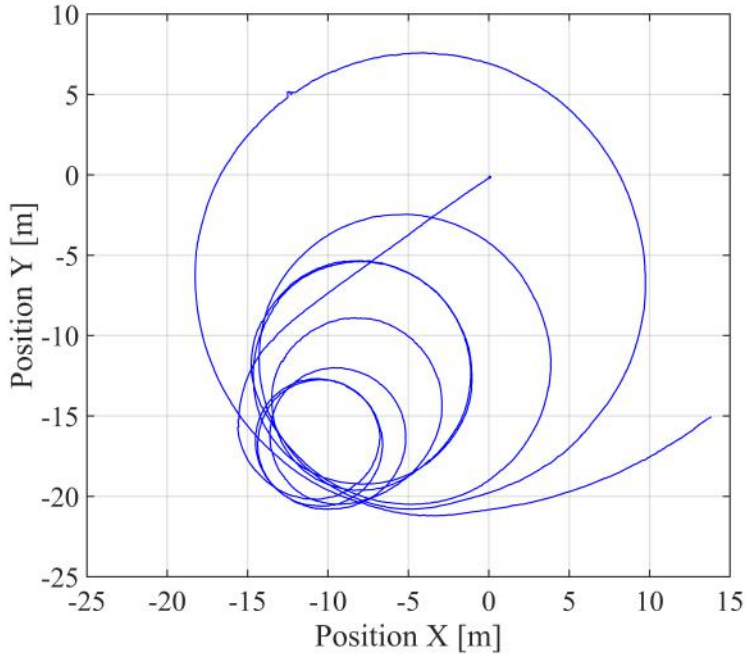


Figure 4.15: Position of the testing vehicle during test with snow chain mode off.

From this test it was found that the steering angle of the rear axle is equal 3° or −3° for every steering angle on the steering wheel.

Maximal steering angle

Every vehicle has the maximal steering range, and it depends on the geometry of a vehicle. For our purposes this geometric steering range is irrelevant, but when vehicle moves faster it is harder to turn the steering wheel (for same reason lookup tables are used for P and I constants in regulator) and because the test vehicle has limitation of the maximal steering moment, we can find the maximal steering angle for this moment. For purposes of this test, maximal moment was set on the steering wheel and vehicle started accelerating. As can be seen in the figure 4.16.

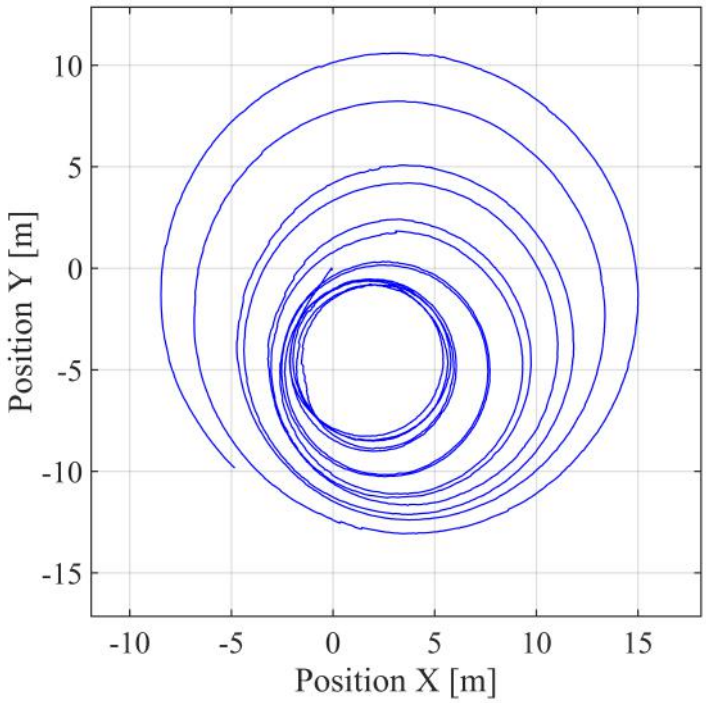


Figure 4.16: Maximal steering test with constant value of moment.

From this, we can measure the minimum diameter for a set speed, and then find the dependency of the steering angle on the speed, which can be seen in the figure4.17.

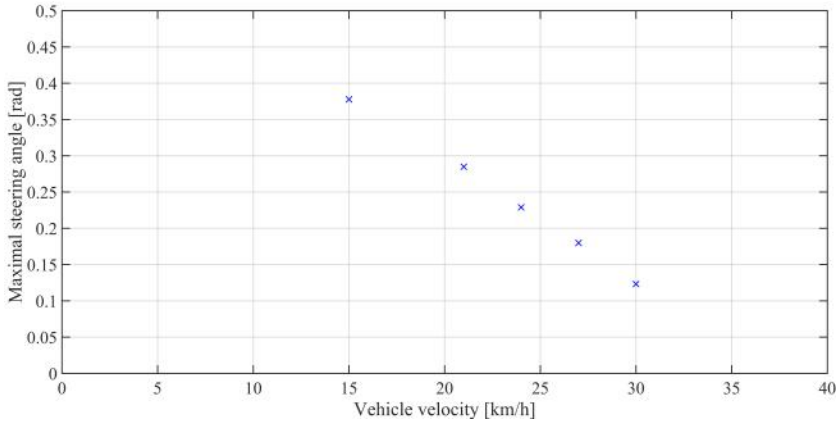


Figure 4.17: Steering angle dependency on vehicle velocity.

From this graph we can see that the maximal steering angle is high enough, for our speeds not to limit our maneuverability.

Tire stiffness coefficient

The last parameter that is needed is the tire stiffness coefficient which represents the amount of force that can be transferred on the tarmac. These coefficients are posing in the dynamic model 2.3.3. This means that the test has to be done in higher but still constant speeds, it is also important that the test vehicle is not in a skid. To set the proper value of this coefficient Matlab parameters estimation toolbox was used, where inputs to this toolbox were our dynamic model with yaw rate $\dot{\psi}$ of the test vehicle, which was again measured by vehicles control units. Changeable parameter was front tire stiffness coefficients C_f.

Results can be seen in the figure 4.18, where the model with C_f represents reality best and therefore was selected.

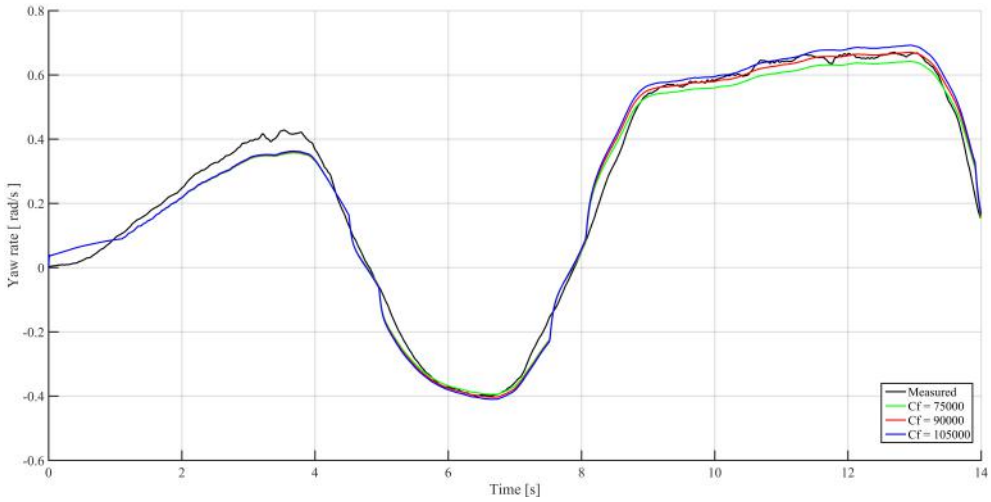


Figure 4.18: Front tire stiffness coefficient estimation.

4.4. Parameter validation

Before we can assume that our designed values are correct some validation is needed. For our purposes we can use more than one criteria of validation. Therefore we will use a control unit with a sensor of yaw rate as the first criteria, and differential GPS as the second criteria of validation.

The test driver will make sets of manoeuvres with the test vehicle in slow speeds with and without the snow chain mode on to validate the kinematic model and then he will do some manoeuvres in higher speeds again with and without snow chain mode to validate the dynamic model.

As the test vehicle is measuring every input of the driver, these inputs can be then set as inputs of our models to evaluate their correctness. Two of these validation can

be found in the figure 4.19 and 4.20. Data from other tests can be found in the list of attachments in the chapter 6.

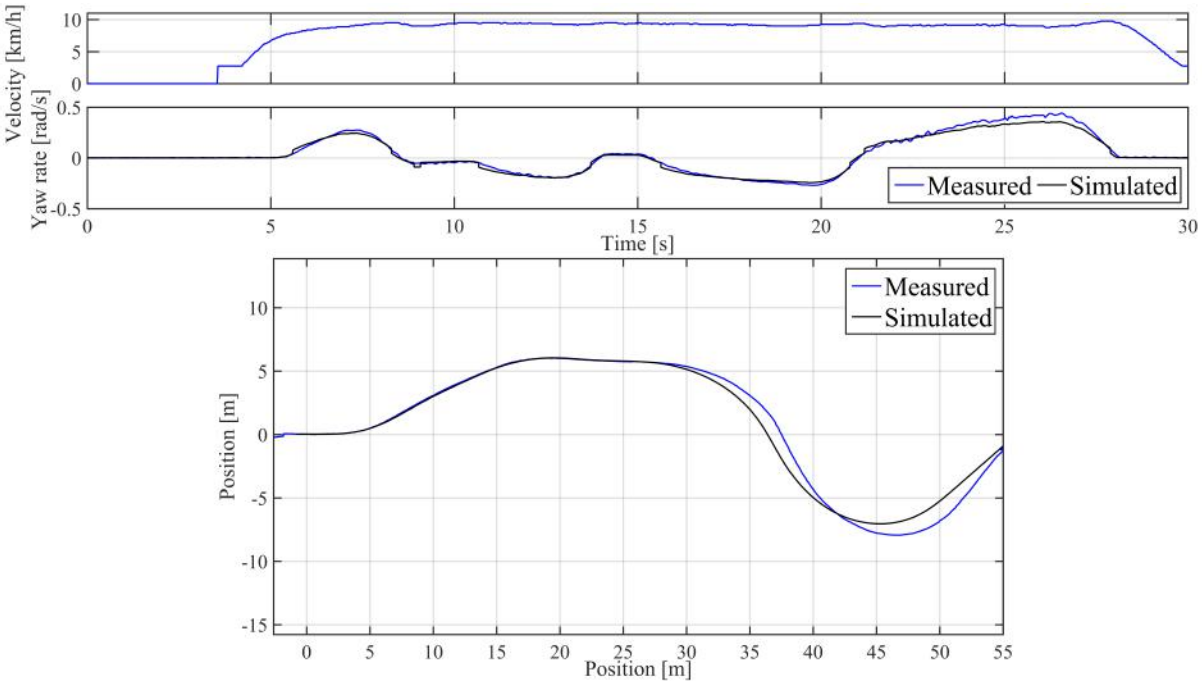


Figure 4.19: Validation of kinematic model with snow chain mode off.

The measured and simulated yaw rate in the figure 4.19 is nearly the same, but these small errors will appear as more significant errors in position of a car due to double integration that has to be done for calculation of location of the vehicle, as it can be seen in equations 2.18, 2.19 and 2.3.

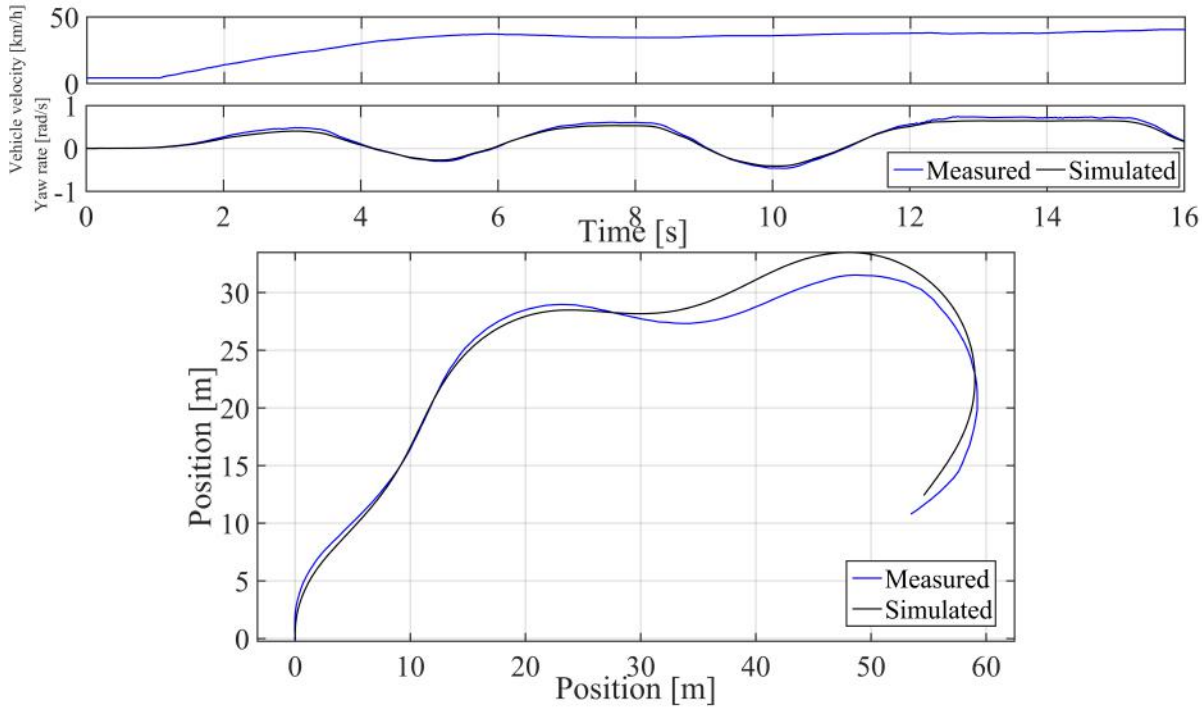


Figure 4.20: Validation of dynamic model with snow chain mode off.

For the dynamic model in the figure 4.20 we can see the same position inaccuracy even though the yaw rate simulated values correspond more with the measured ones. This inaccuracy is caused by the double integration again and by the initial condition of the simulation. Because we can not start the simulation from zero speed, because of the division by the speed in the dynamic model we need to start the simulation at a speed higher than $2,5\ kmh^{-1}$ because the test vehicle is not capable of measured speed below this value. This will be resolved in the section 4.5 as the switching between the kinematic and dynamic model will be introduced.

4.5. The vehicle simulation

Because of the limited time that can be spend with the testing on the testing track, it is needed to verify that the vehicle control works properly before it is tested on real vehicle. Testing simulation model in Simulink was build for these purposes. In this section I will introduce Simulation model, simulation that it was used for and control mechanisms that will be used in real testing vehicle.

4.5.1. Simulink model

To simulate behaviour of the test vehicle Simulink model from equations in section 2.3 was made, and can be seen in figure 4.21. This part of the model was verified in chapter 4.4.

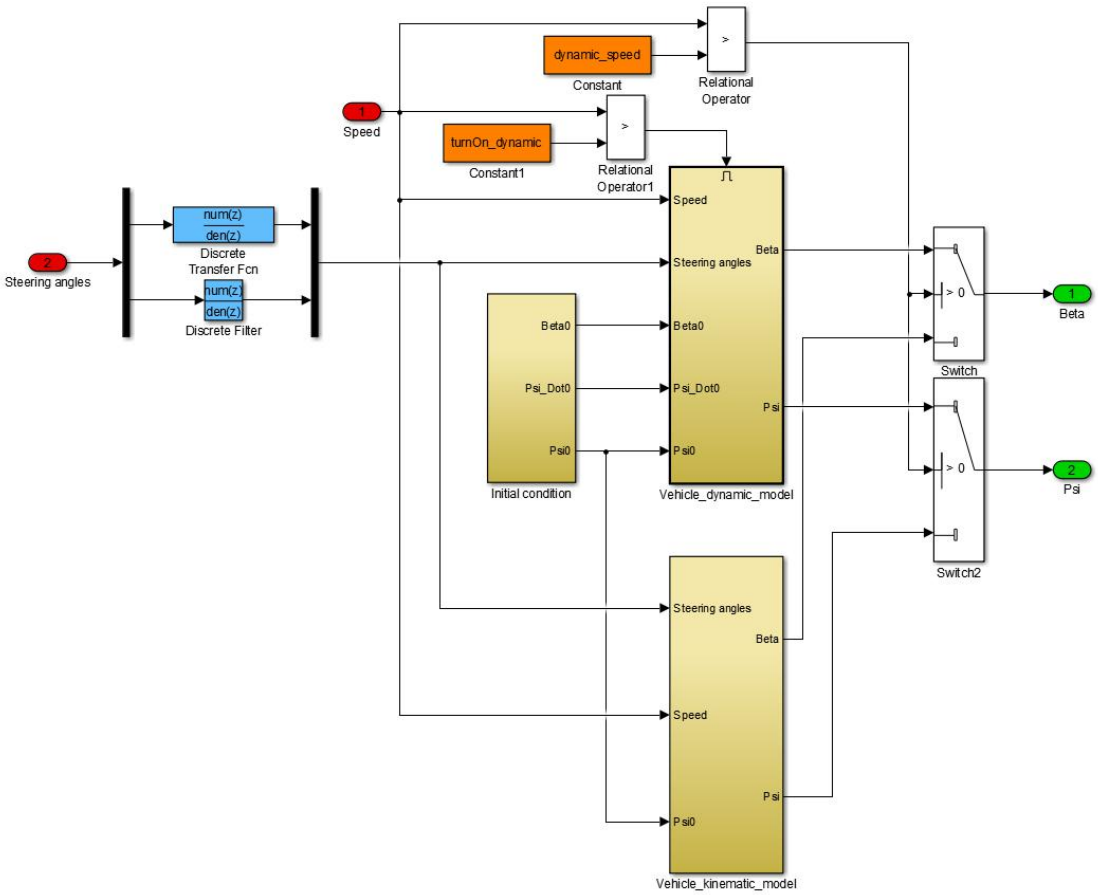


Figure 4.21: Simulink model which represent behaviour of the test vehicle

Switch between kinematic and dynamic model was implemented to the model, this switch is controlled by speed. Furthermore activation of the dynamic part of the model had to be added because for speed equal zero are outputs approaching infinity. Input, output (measurement) and process noises were added to make model more realistic. Transfer functions on steering angle were estimated from measurements of front wheels on step responses of variant heights. These measurements can be found in the list of attachments in chapter 6. Transfer functions represents response of PID regulator and mechanics of front/rear steering system.

Next added to the model is a feedforward algorithm with PID feedback, which will take care of calculation of steering angles needed to drive through the test track, for this purposes we calculated and saved data needed to drive through this track. Furthermore algorithm was designed to calculate steering angle for dynamic model and for kinematic model. As it can be seen in figure 4.22 switches to change calculation for different types models were added.

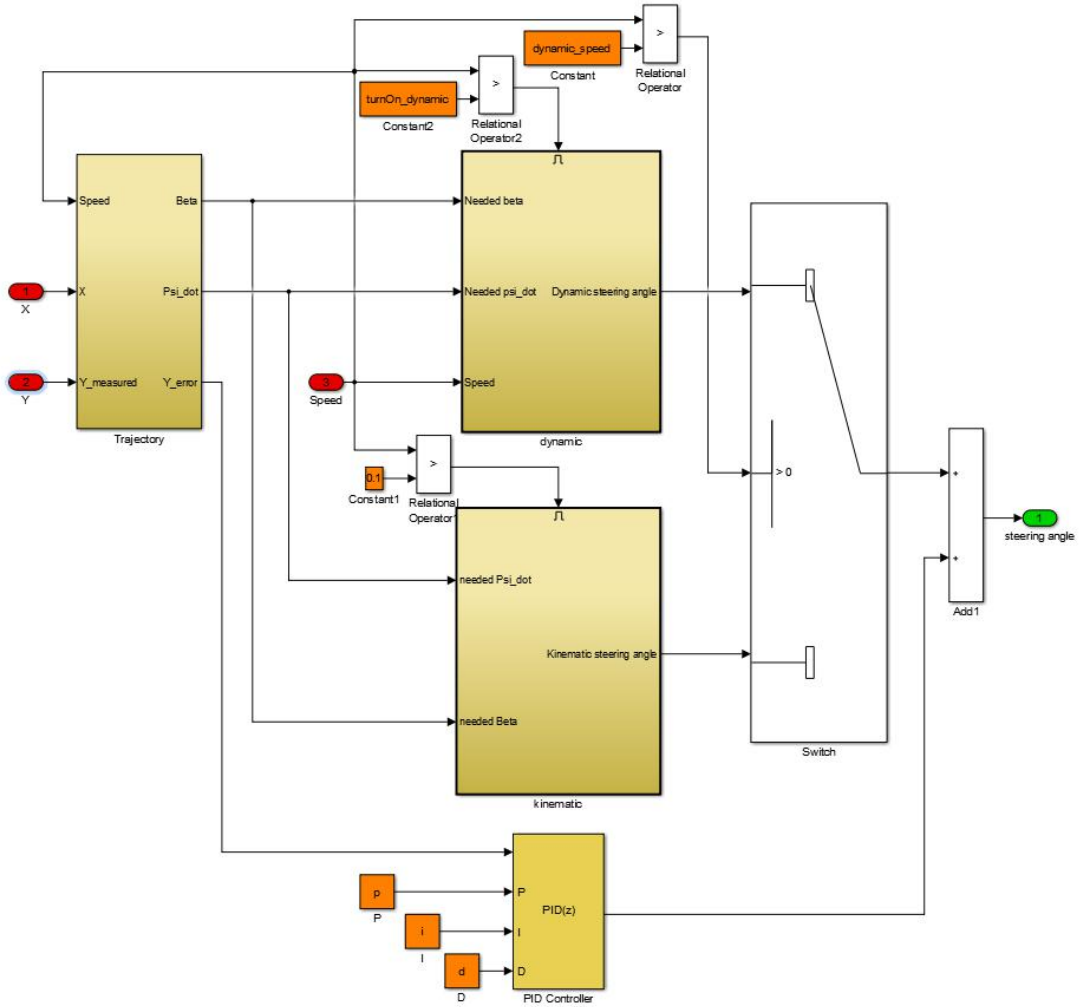


Figure 4.22: Feedforward algorithm for the test vehicle.

Algorithm present inside Dynamic and Kinematic block are shown in figure 4.35 and 4.34 respectively.

Next added is a model of the vehicle without any noise to work as a observer. Kalman filter was added too, to determine most precise position of the vehicle. Furthermore steering angle preprocessor was added to prepare output of Feedforward algorithm for usage in the Simulink model.

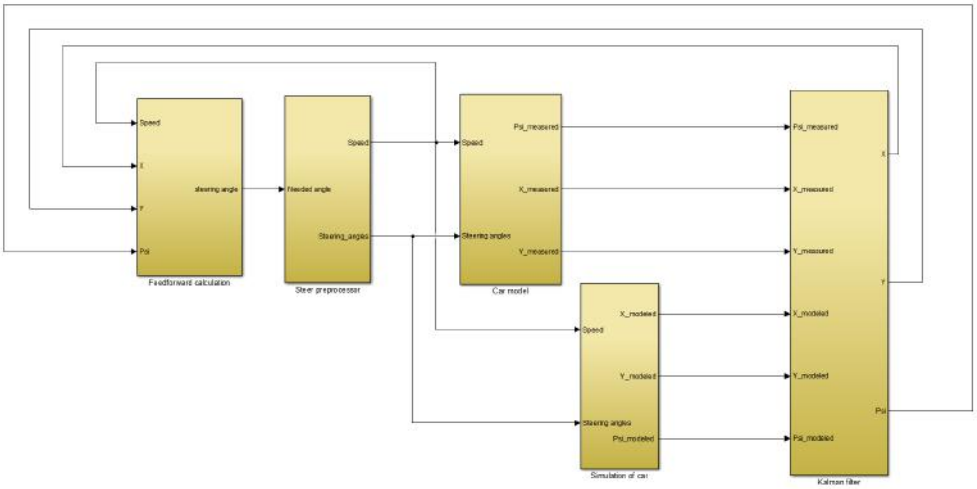


Figure 4.23: Model of the vehicle used for simulations.

4.5.2. Test manoeuvre

In the assignment of this thesis is specified that the control test has to be moose test. The test was normalized by International Standard Organization and the Association of the German Automotive Industry, and its name was changed to ISO 3888-2 [14]. Test is mainly used for evaluation of the handling performance of a vehicle. Based on 3 cone lanes with a total length 61 meters a double lane change is defined, which must be completed with maximum speed.

Test procedure

The ISO double lane change consists of an entry and an exit lane with a length of 11 m. The width of all lanes are dependent on the width of the test vehicle. The lateral offset between entry and side lane is 1 m and longitudinal offset is 13.5 m. For the same lateral offset the side lane exit is slightly shorter longitudinal displacement of 12.5 m. 2 metres after the start of the entry lane the throttle should be released, but for our lower speed we will try to hold constant speed to even get to the end of manoeuvre.

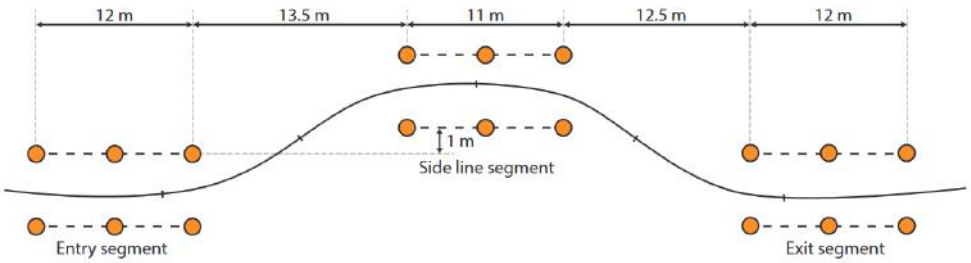


Figure 4.24: Double lane change test, ISO 3888-2.[14]

4.5.3. Simulations

This section is dedicated for simulations of vehicle with different controllers because of limited time that can be spend with the test vehicle and the test drivers.

Feedforward

First to validate is the Feedforward algorithm, this is done by turning off the Feedback. Trajectory and results can be seen in figure 4.25. This test was done for low vehicle velocity to make sure, that we are in stable speeds.

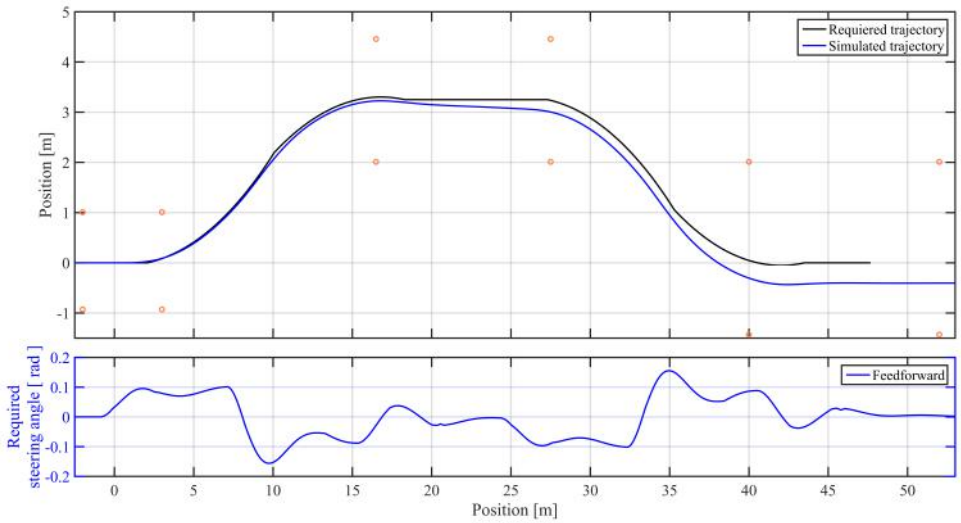


Figure 4.25: Simulated vehicle trajectory for feedforward controller.

Red circles in graph marks positions of cones on the test track, these positions was calculated from [14]. Black line is required trajectory of the CG of the test vehicle and blue line is trajectory that was calculated with help of feedforward algorithm and simulated vehicle.

Feedback

Problem with only Feedback PID controller of steering in vehicle is that controller will react only on error in the position of the vehicle. Which practically means, that the vehicle will never be on position that it is supposed to be. To eliminate this error, method that take into account the speed of setting desired steering angle and speed of the vehicle will be used. This can be only used when the trajectory is known before the steering starts. For our purposes, velocity of the vehicle is measured and than taken into account, when calculating reposition of the trajectory. This reposition is made so the position of the vehicle is not shifted to the position of the original trajectory. Results can be seen in figure 4.26, where is shown original trajectory, shifted trajectory and position of simulated vehicle.

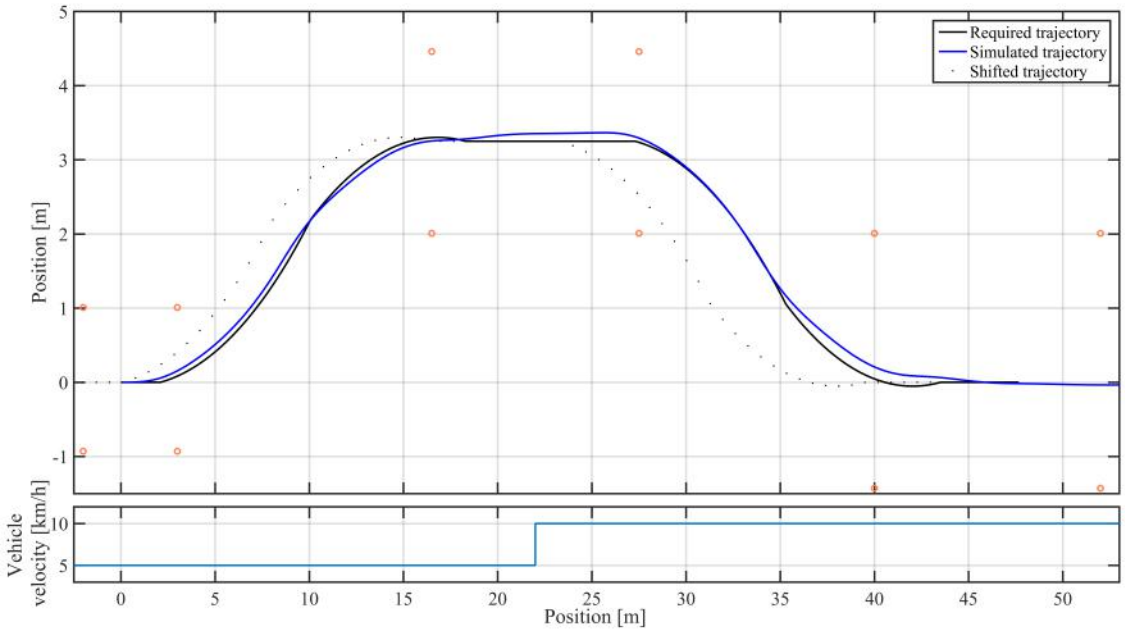


Figure 4.26: Simulated vehicle trajectory for feedback controller.

Feedforward and Feedback

Combination of Feedback and Feedforward controller should ease work of the Feedback algorithm by calculating required steering angles before there is an error in position. In this situation, requested position on the Y-axis is not shifted, because the Feedback controller has to make adjustments which are relatively small. Results can be seen in figure 4.27, together with the requested steering angle form Feedforward and Feedback control.

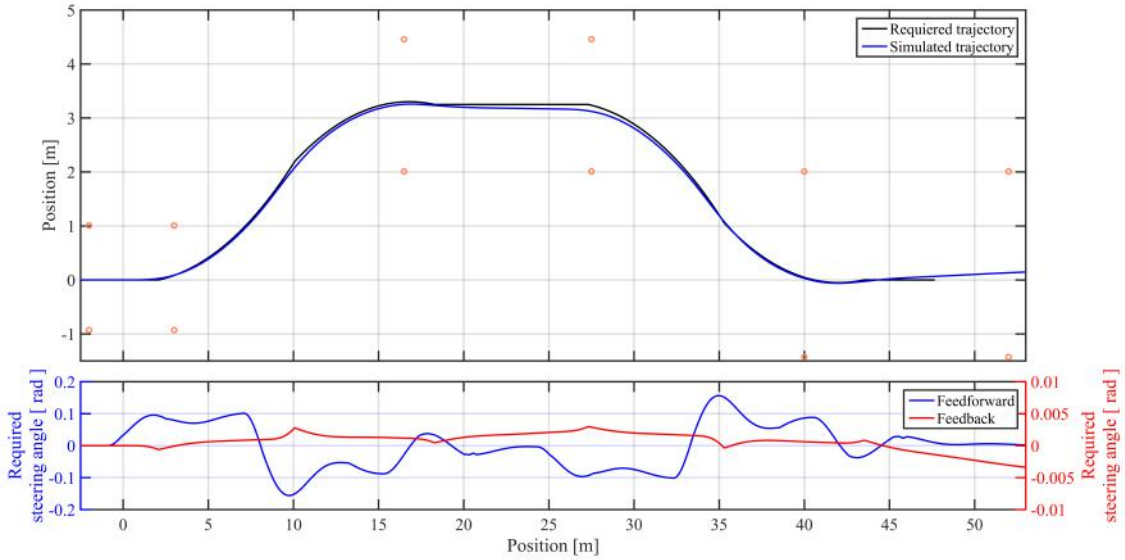


Figure 4.27: Simulated vehicle trajectory for feedforward and feedback controller combined.

Real limited steering in higher speeds

Because of the maximum moment limitation, and maximum value of derivation of this moment it is sure,the maximal speed of the real vehicle which can drive through the manoeuvre is lower than it is specified in assignment of this thesis. However maximal speed that can be used to drive through manoeuvre can be identified in simulations and later verified in test. In figure 4.28 can be seen simulations for various speeds of the test vehicle.

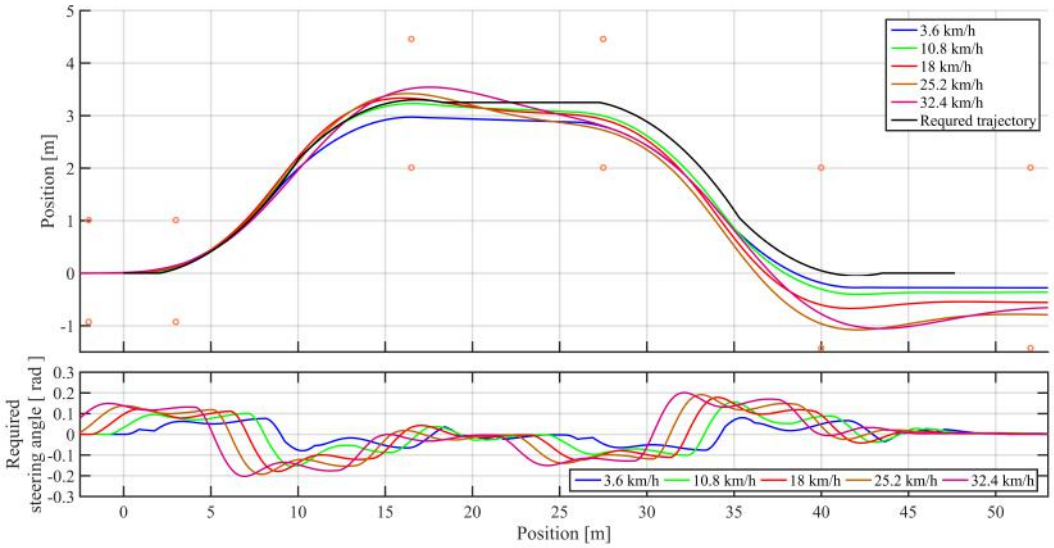


Figure 4.28: Simulation in various speed for simulations of real test vehicle.

With the known width of the vehicle, which is $1.937m$ we can determine, which vehicle speed will overturn cones on the test track. Speeds of $25.2km/h$ and $32.4km/h$ will hit cones that are shown in figure 4.29. Another problem with higher speed is the start of the manoeuvre, in figure 4.28 can be seen that the output of the regulator starts before the manoeuvre starts.

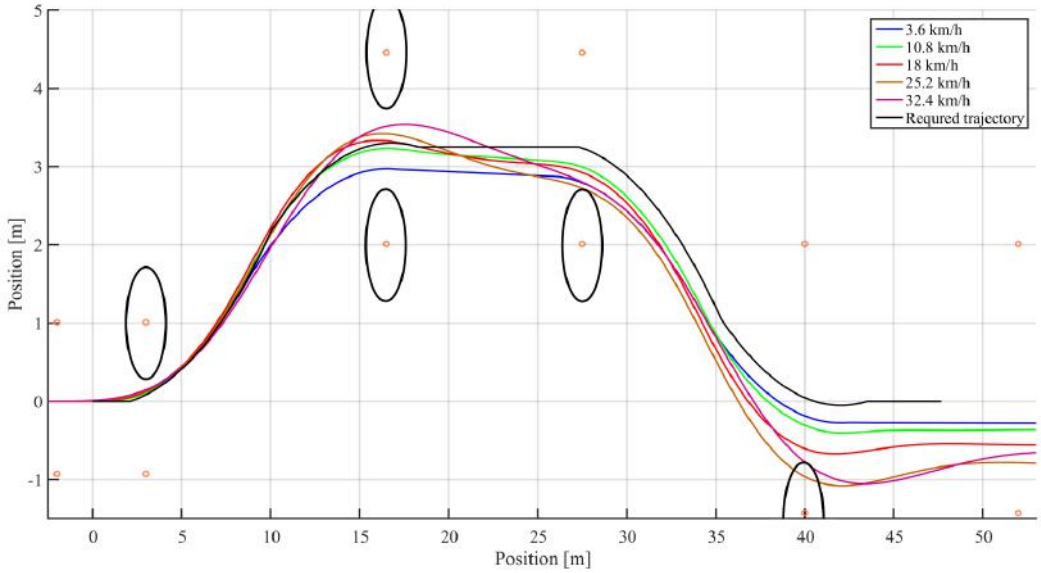


Figure 4.29: Evaluation of the position in the test.

Where the circles around the cones on the graph are ellipses due to scale of the axis. Intersection of these ellipses with the trajectory of the vehicle means that the given cone was hit and thus that the vehicle failed the test.

Ideal steering in higher speeds

With the ideal steering, there are no limitations of the moment on steering wheel or any limitations of derivation of this moment, which means that the turning speed of steering wheel is limited very little. Limitation of the steering wheel turning speed will be in Simulink represented by rate limiter.

In the figure 4.30, can be seen simulations of the vehicle in the speed of $18km/h$ as a comparison with real simulated vehicle, and for speed of $54km/h$.

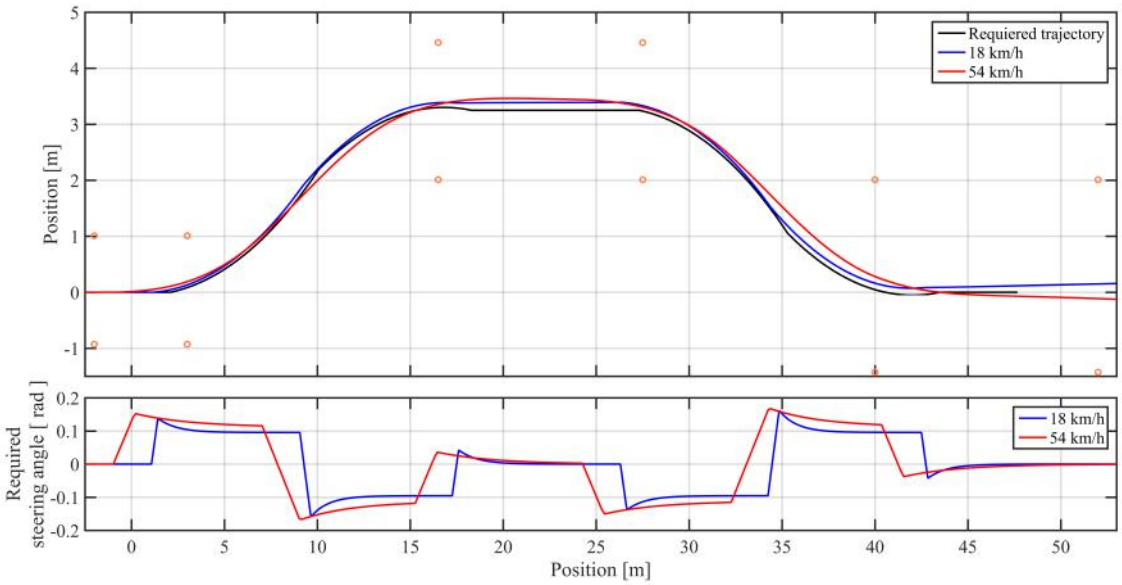


Figure 4.30: Ideal steering vehicle model simulations for speed of $18km/h$ and $54km/h$

4.6. Experimental models

Final versions of models, which are going to be tested on the test vehicle are introduced in this section.

4.6.1. Feedback control model

In the section 4.2.2, the PID regulator was introduced and his implementation in control model, which can be seen in figure 4.31.

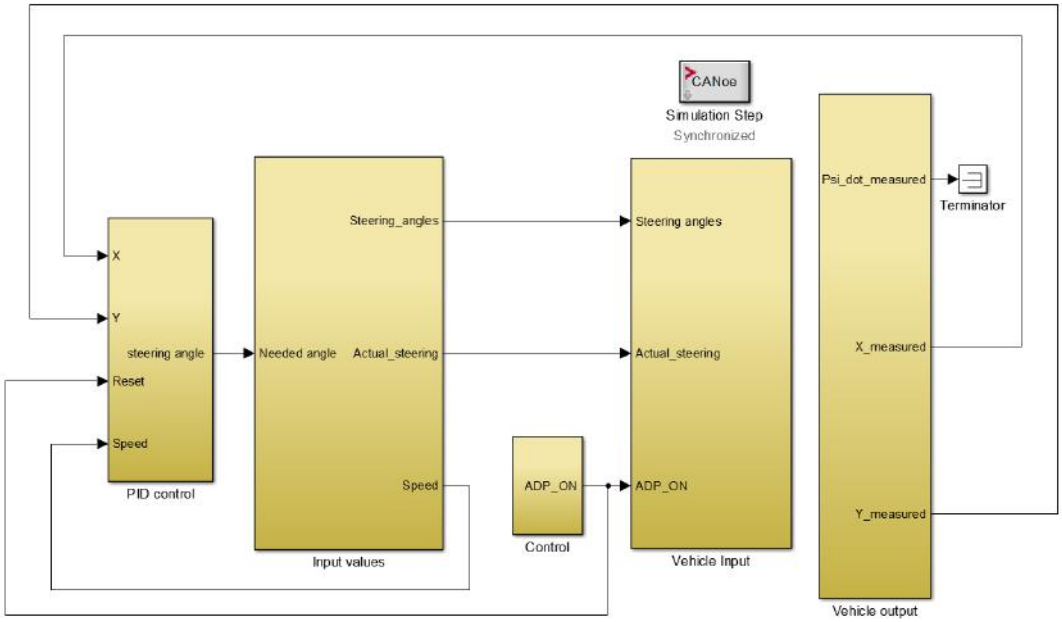


Figure 4.31: Simulink model for the test of position feedback controller.

First block from left in Simulink is PID controller, detail of this block is shown in figure 4.32. Block named Input values contain output communication of the test vehicle (actual speed of the vehicle, Actual steering angle of front wheels), and post process of required front wheel steering angle, which consist of adding rear wheel steering angle. Control block serves only as control input, to turn on and off manoeuvre. Block Vehicle input contains PID regulator, and input communication (moment on steering wheels). First block from the right contains outputs of the differential GPS.

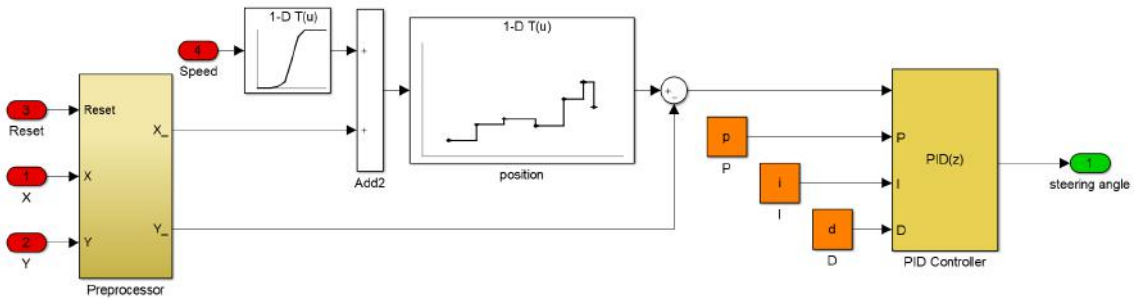


Figure 4.32: The PID controller for the test model.

The first Simulink block in figure 4.32 after the inputs is the Preprocessor. This block prepares values of position from GPS for our regulator. Which means rotating and translating data to origin, to simplify further controls. Lookup table in this model gives ideal Y position for given X position. This value is shifted by another lookup table which takes into account speed of the vehicle. Last block is the PID regulator.

4.6.2. Feedforward control model

In the section 2.3, the kinematic model and dynamic model was introduced. These models were used to develop Feedforward calculations to control the test vehicle. Simulink control model is shown in figure 4.33

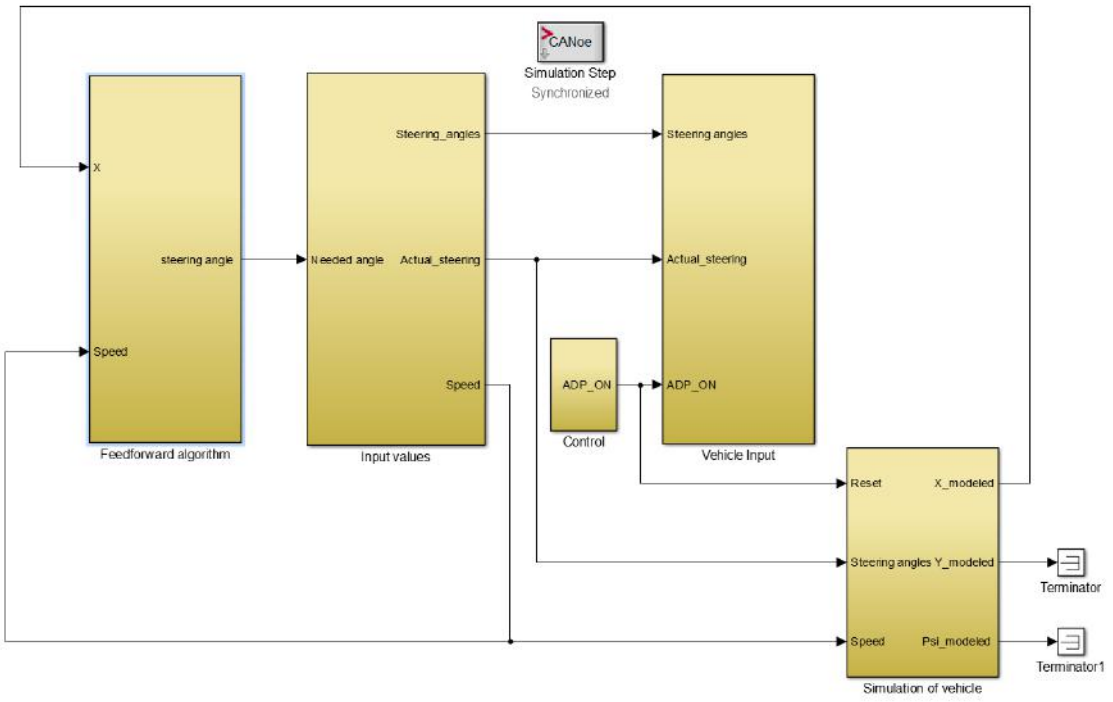


Figure 4.33: Simulink model for the test of feedforward controller.

First block from the left in Simulink is the Feedforward algorithm. Which contains two algorithms to calculate steering angles, both can be seen in figure in 4.34 and 4.35. Second block from left takes cares of output communication from the test vehicle. And is exactly the same as in previous model. Block Vehicle input is again same as in previous model. Last block in the model is Simulation of vehicle, which takes cares of simulating vehicle behaviour, this is done with the equations from section 2.3, and outputs from this model are used in further calculations. GPS data will be used only for evaluation results and not for online controller.

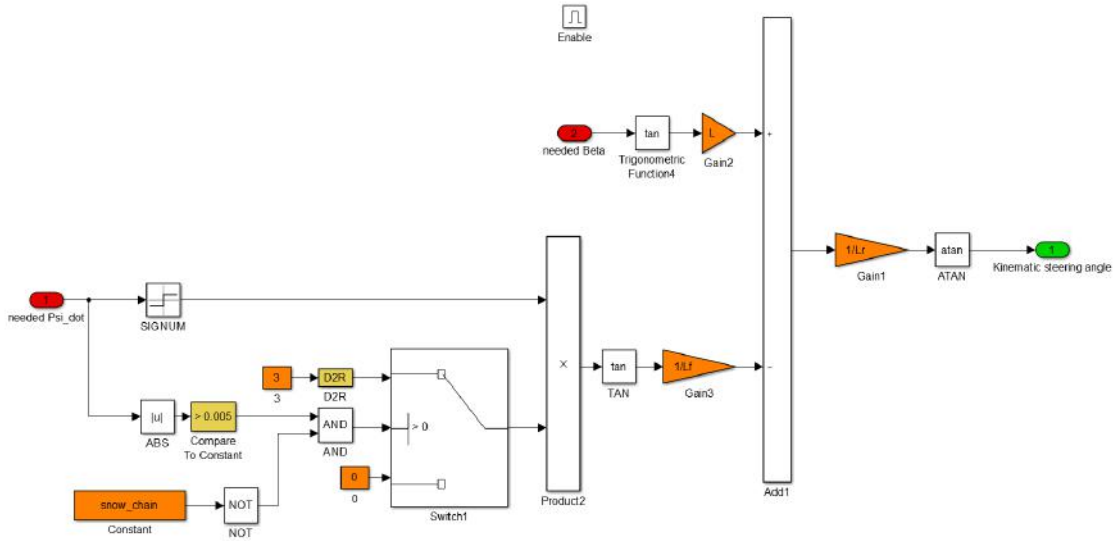


Figure 4.34: Calculation of steering angle for slower speeds.

Kinematic steering angle calculation can be seen in figure 4.34, where is taken in consideration rear wheel steering angle when the snow chain mode is turned off. This equation was created from kinematic model equations.

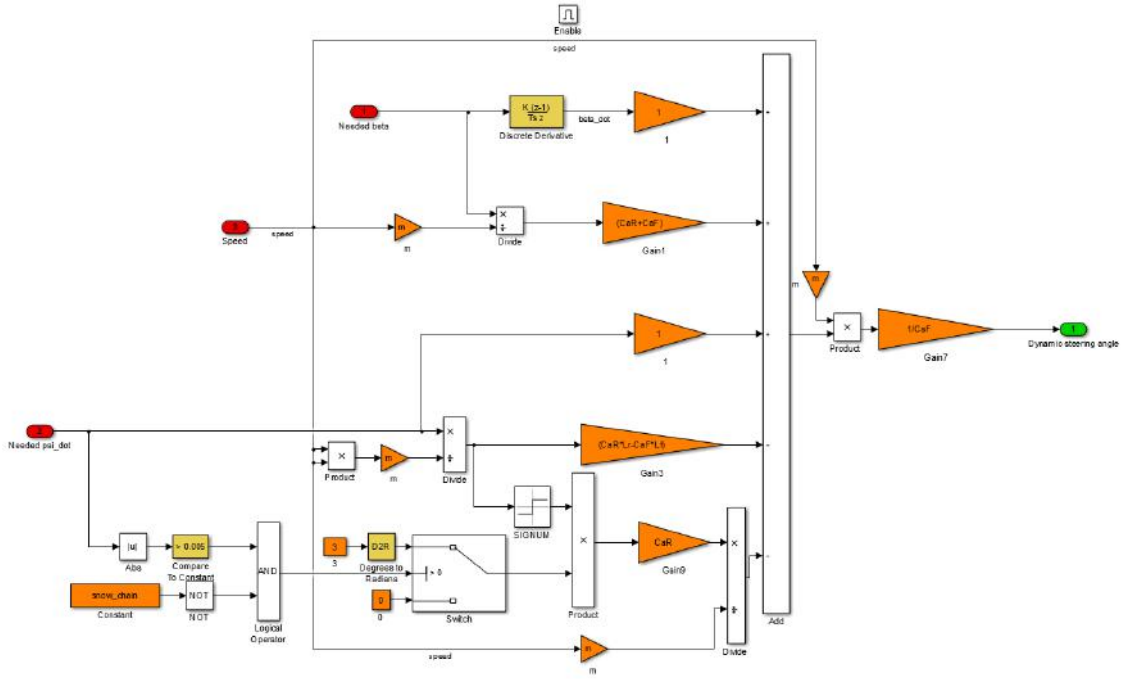


Figure 4.35: Calculation of steering angle for higher speeds.

Dynamic steering angle calculation is shown in figure 4.35. This equation was created from dynamic model equations in section 2.3. Which takes in consideration rear wheel steering angle and dynamic influences of the vehicle.

4.6.3. Feedforward and feedback control model

By combination of the previous controllers we get controller shown in figure 4.36.

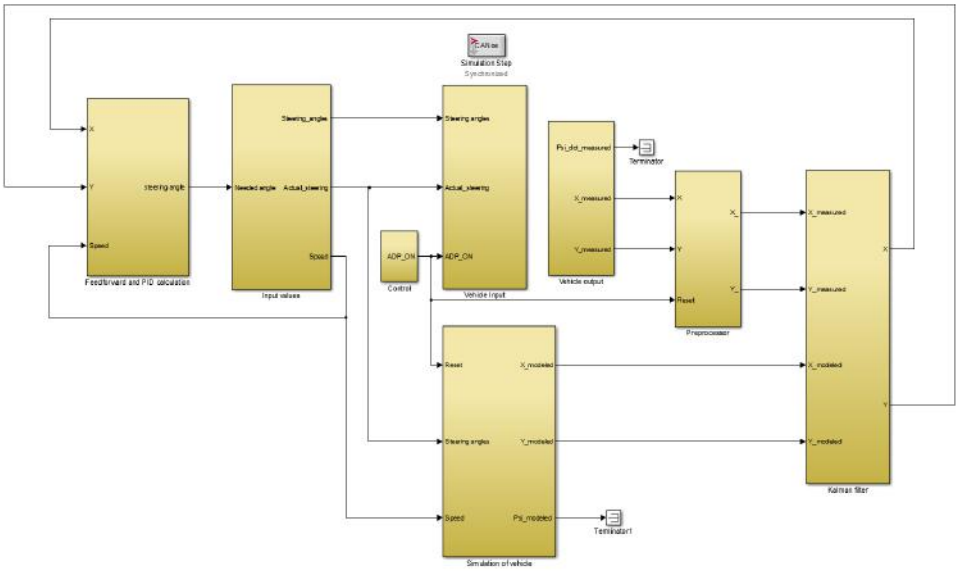


Figure 4.36: Simulink model for the test of feedforward and feedback controller.

First block from the left contain Feedforward and PID controller. And another change from both previous model is Kalman filter which purpose is to calculate more precise position of the test vehicle. Controller of the test vehicle is shown in figure 4.37.

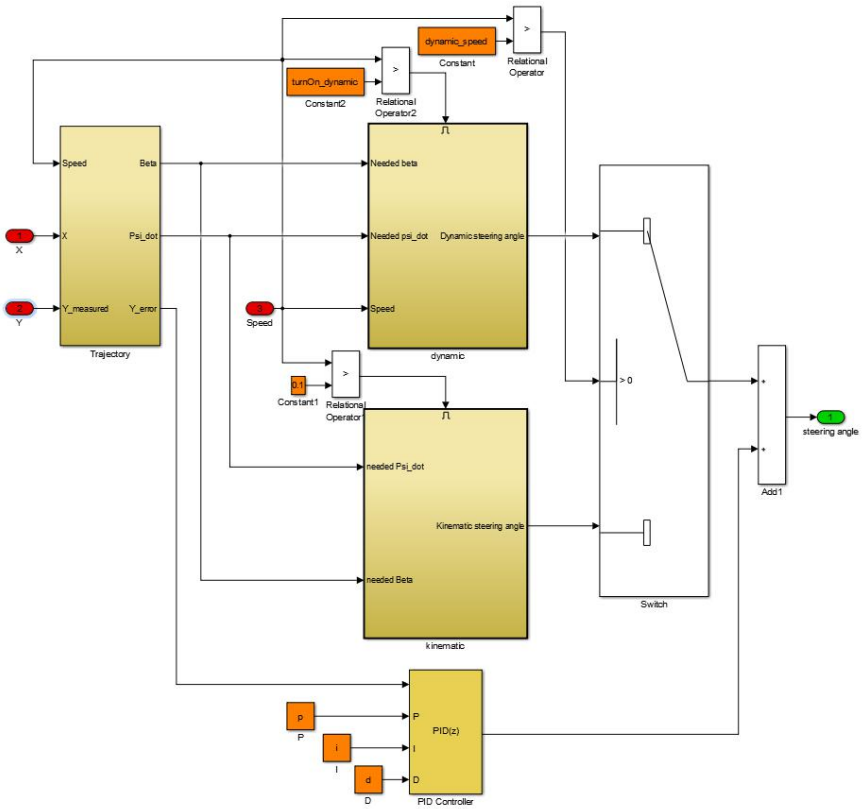


Figure 4.37: Design of feedforward and feedback controller.

First from the left is shown block Trajectory, which makes reference values for PID controller and feedforward controller, than there are two variants of calculations of steering angle and switch to switch between these two variants. This switching depends on speed of the vehicle. PID controller interference is than added together with the Feedforward controller output.

Yaw rate feedback

Because of lack of the precise GPS on most of the vehicle nowadays, one more model was designed. This model combine feedforward control algorithm introduced in 4.6.2 and feedback regulator, but in this case output of the vehicle is Yaw rate (rad/s). Test model can be seen in figure 4.38.

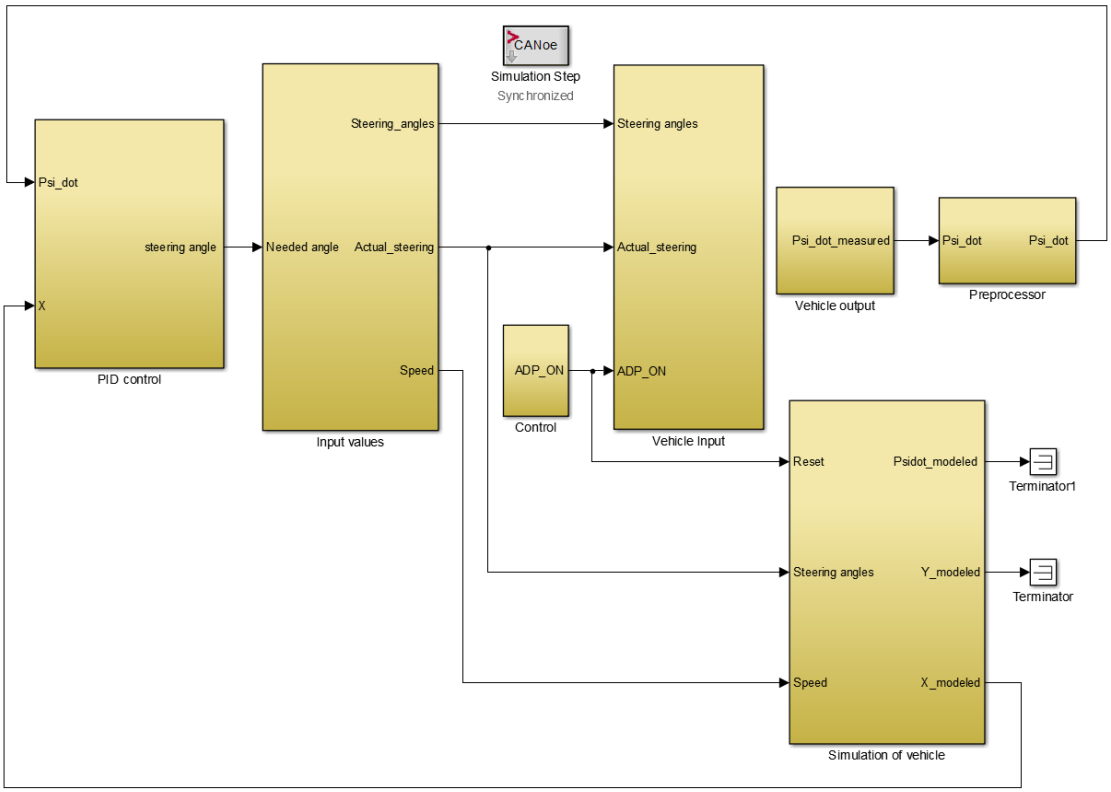


Figure 4.38: Simulink model for the test of feedforward and yaw rate feedback controller.

5. Experimental results

This chapter is dedicated to results of experiments with the real test vehicle.

5.1. Position feedback

First to test is the Feedback controller, which can be seen in figure 4.31. This variant uses position from differential GPS as feedback. To compensate slow speed of the steering wheel trajectory is shifted by value, which was experimentally determined by series of test in various speeds.

5.1.1. Results

The test results can be seen in figure 5.1. Speed of the vehicle was controlled by the driver and value was $10km/h$.

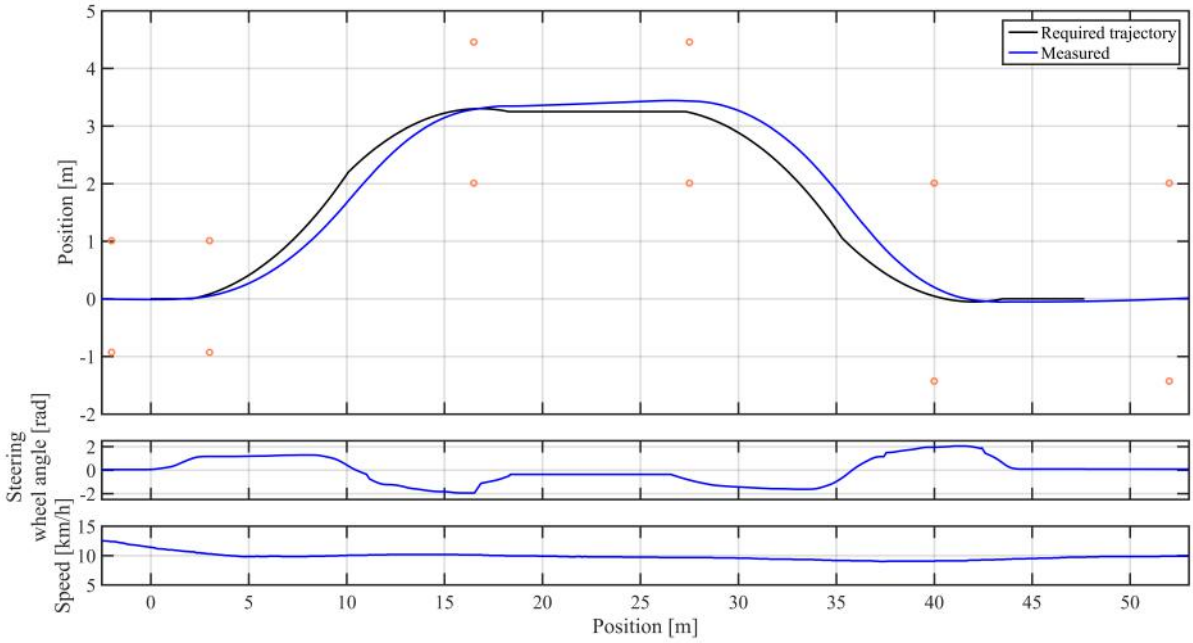


Figure 5.1: Test results for feedback controller at speed of $10km/h$.

As can be seen test vehicle passed the test manoeuvre, but required trajectory together with the measured are not completely same, this can be due to wrongly set shift of the manoeuvre and/or by inconsistency in speed of the vehicle.

5.2. Feedforward

Next variant to test is Feedforward control, this will be done without any feedback from the real vehicle. Speed of the vehicle is in test again controlled by the driver and is assumed that it is constant with value of $10km/h$. Controller used in this test, was introduced in section 4.6.2.

5.2.1. Results

Series of tests was made for this controller. The final version of algorithm can be seen in section 4.6.2. Position measured from GPS is used only for offline evaluation of the test and not for the control.

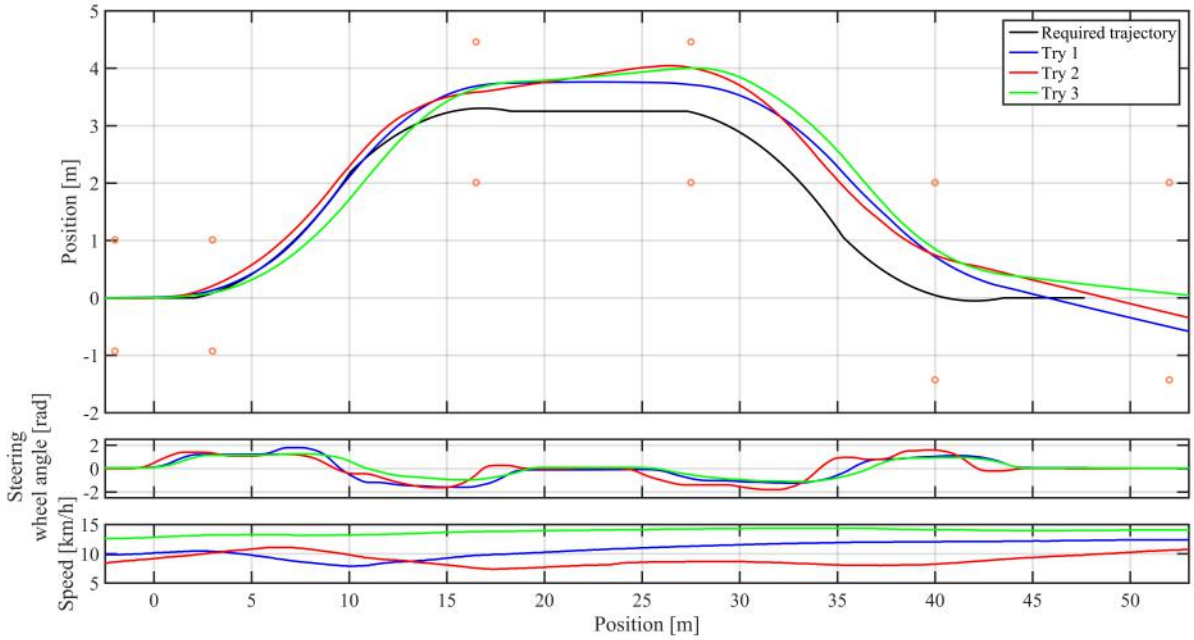


Figure 5.2: Test results for feedforward controller at speed of $10km/h$.

Required and measured trajectory are not same in any of the measurement, but all version were able to pass the test manoeuvre without knocking over any of the cones. Problem with this control algorithm is that any roughness on the tarmac can change the course of the vehicle, and without any feedback will not make any corrections. The overshoots of the position in the second corner and non zero value of heading of the vehicle is due to wrongly identified transfer function of the front wheel. Identified transfer function is faster than the real one and this difference make error in position.

5.3. Feedforward and position feedback

Combination of previous two controllers should give best results, the test model is introduced in section 4.6.3. Position from the GPS is used as feedback value together with position calculated from mathematical model. Mathematical model calculations are included because of the slow rate of sending data from GPS. Data from GPS are also used for evaluation of the test.

5.3.1. Low speed results

First test was made at the speed of $10km/h$.The test results can be seen in figure 5.3

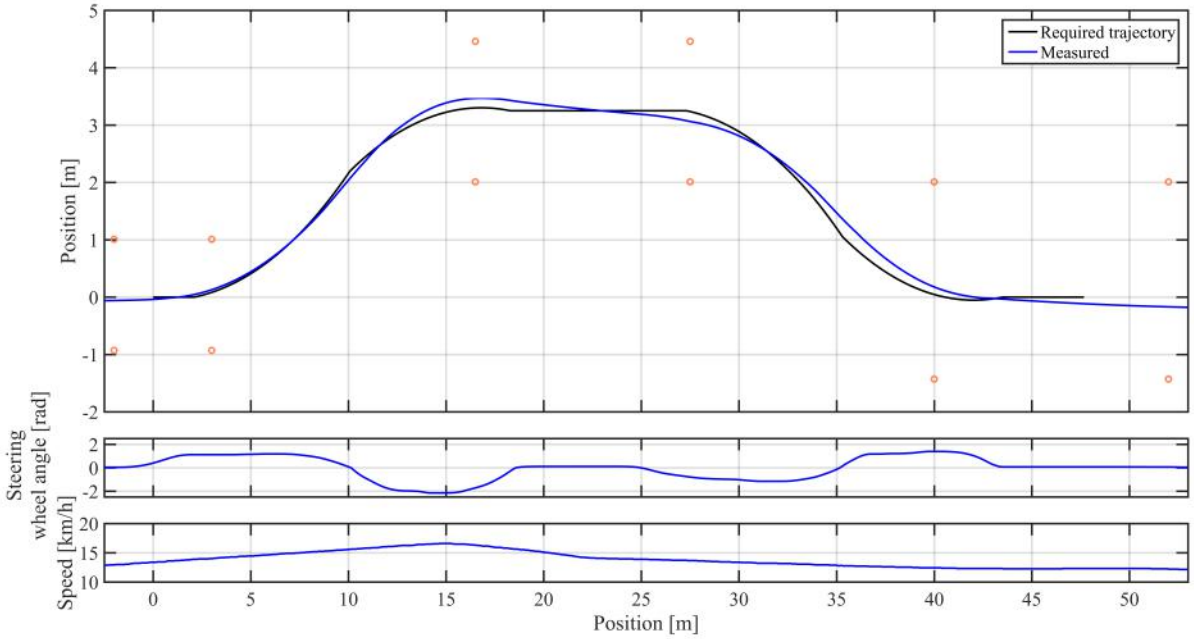


Figure 5.3: Test results for feedforward and position feedback controller at speed of $10km/h$.

Maximal error from required value in this test was $0.2m$ which is low enough to pass the test manoeuvre.

5.3.2. High speed results

This test was done for speed higher than $20km/h$, this speed was chosen based in simulations in section **??**.

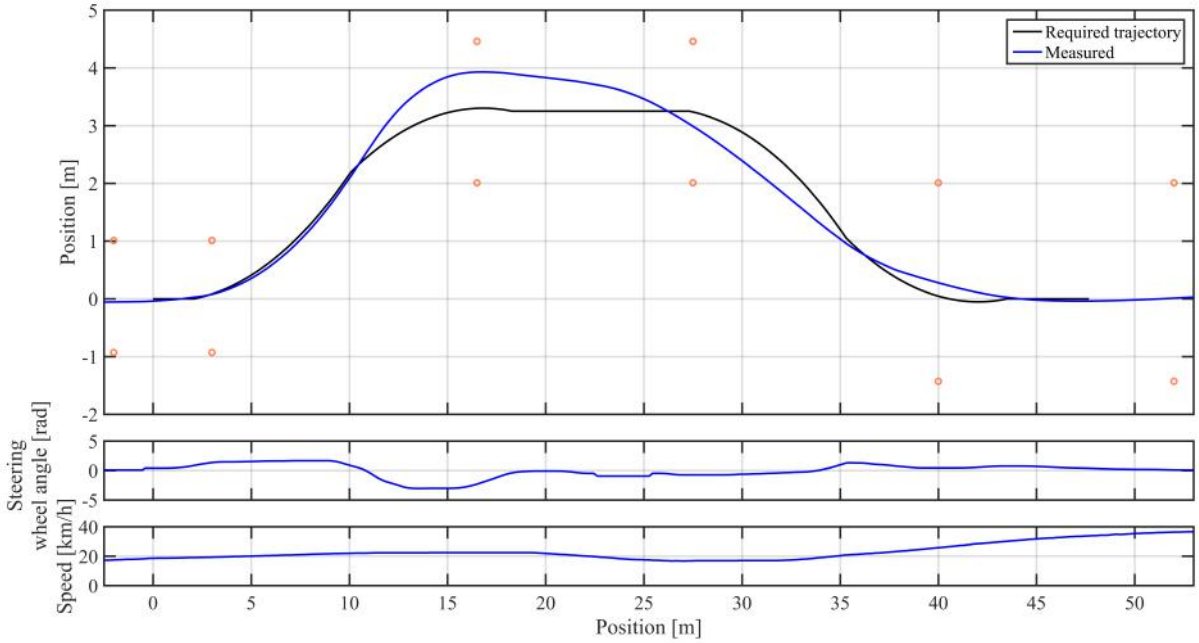


Figure 5.4: Test results for feedforward and position feedback controller at speed of $20km/h$.

At this speed the test vehicle failed to pass the manoeuvre, this difference between simulations and real test vehicle can be due to wrong estimations of front wheel steering mechanism.

5.3.3. Yaw rate feedback

Previous feedback control does not work without the GPS which means, that in places with no signal or when the test vehicle is not equipped with one, control mechanism will not work. As a solution feedback controller which depends only on vehicle implemented sensor was designed and introduced in section 4.6.3. Test speed of the test vehicle in the test was $10km/h$ and results can be seen in figure 5.5.

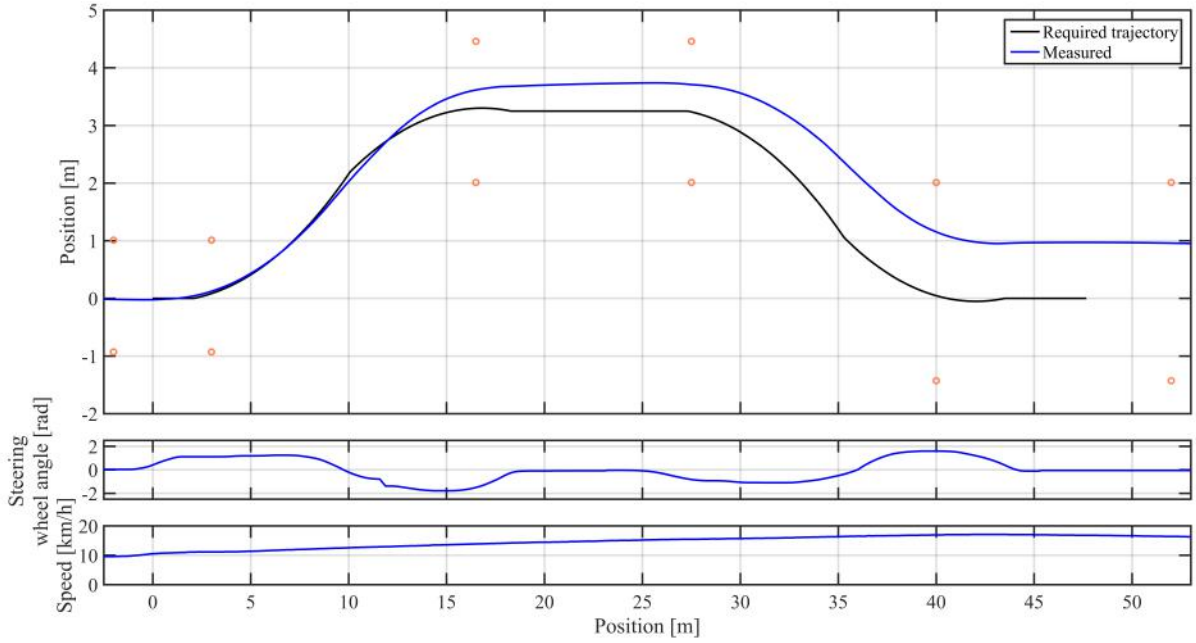


Figure 5.5: Test results for feedforward and yaw rate feedback controller at speed of $10km/h$.

From the results can be seen that the feedback controller improved error in heading of the vehicle, but position error cannot be reduced by this controller. This is probably again due to wrongly estimated transfer function of front steering wheels.

5.4. Evaluation of the tests

All the test were impaired by the inconsistency of speed. This is due to combination of more factors. First is that the speed of the vehicle is controlled by the driver, this is because vehicles speed assistant works only from $30km/h$, second is that the driver could not use brake to control speed, this is due to safety reasons as the brake pedal stops any controller. Third is that the lowest stable speed of the vehicle only $5km/h$. And last problem is in the temperament of the vehicle, because the test vehicle is sport car any input on throttle is a significant change of the speed.

6. Conclusion and future work

To achieve the main goal it was needed to get acquainted with the vehicle network architecture, vehicle dynamics, kinematics and electric power steering. After the theoretical part of the thesis, the vehicle network was adjusted to fit our requirements. When the new network topology was finished, the signal gateway, which ensures the normal function of the vehicle, was written. The parameters of the vehicle which are needed for simulation purposes and for the controller, were measured or estimated from tests. In a preparation for the main final tests a set of models was made.

The main objective of this thesis is to develop an algorithm, which will be able to drive the testing manoeuvre at a minimal speed of 50 km/h. This has not been reached due to the manufacturing and safety limitations of the steering mechanism on the test vehicle. However, the steering controller has been tested at simulations of the test vehicle, both with the real and ideal steering mechanism. The simulation with the real steering controller has confirmed our concerns about the limited maximal speed, at which the vehicle can pass the test. And the simulations with the ideal steering have proved that the steering controller is able to pass the test at speeds higher than 50 km/h.

Simulations of the vehicle have been supported by the tests with the real test vehicle. Where simulations predicted that the maximal speed was around 18 km/h, our test vehicle has passed the test at a velocity of 20 km/h. However, the correct function of the several algorithms has been confirmed. The first is an algorithm with feedforward control, the second is the feedback control where the output value of the vehicle is position, the third is a combination of previous two and the last is a combination of the feedforward algorithm with the feedback control, but in this case the vehicle output value is yaw rate. This has been done to get rid of a necessary usage of differential GPS in the second and third case.

The actual version of the gateway and vehicle steering controller is fixed and approved by Porsche Engineering Services s.r.o.

Conclusion of master's thesis goals

- The electric power steering has been analyzed
- The CAN and FlexRay communication have been analyzed and described
- The infrastructure of the testing vehicle has been analyzed and signal gateway has been programmed
- The feedback steering controller has been designed and tested
- The controller has been designed, tested on simulations and evaluated
- Tests with the real test vehicle have been made and evaluated

The follow-up work can use the established framework and improve the vehicle manoeuvrability by designing a new electronic control unit for the electric power steering and thus remove limitations that are present in the current test vehicle. After this, the steering controller can be verified for higher speeds and modified if needed.

Bibliography

[1] Bohn, C., Atherton D.P.: *An analysis package comparing pid anti-windup strategies.* IEEE Systems Magazine, 34 - 40 p., April 1995.

[2] Bosch: *CAN Specification version 2.0 Overview 5 April, 1995* , accessed 13 April 2019, http://esd.cs.ucr.edu/webres/can20.pdf.

[3] Dr. Ing. h.c. F. Porsche AG.: *Panamera specifications, 2019* , accessed 13 April 2019, https://auto.porsche.cz/modely/panamera/panamera-turbo/technicka-specifikace.

[4] Govender, V., Müller S.: *Modelling and Position Control of an Electric Power Steering System.* IFAC-PapersOnLine, 312 - 318 p.,2016.

[5] Mechanical booster: *How Power Steering System Works Overview December, 2017* , accessed 13 April 2019, https://www.mechanicalbooster.com/2017/12/power-steering-system.html.

[6] MONDEK, M.: *Active torque vectoring system for electric drive vehicles.* [Diploma thesis.] Prag: ČVUT, FEI, 2018. 82 p.

[7] My car does what: *Lane Keeping Assist Overview 2019* , accessed 13 April 2019, https://mycardoeswhat.org/safety-features/lane-keeping-assist/.

[8] National Instruments: *FlexRay Automotive Communication Bus Overview 19 March, 2019* , accessed 13 April 2019, http://www.ni.com/de-de/innovations/white-papers/06/flexray-automotive-communication-bus-overview.html.

[9] Pacejka, H.B.: *Tire and vehicle dynamics 2nd edition.* Butterworth-Heinemann, 2005. 672 p. ISBN 9780080543338.

[10] Electronics Hub: *PID Controller-Working and Tuning Methods Overview December, 2015* , accessed 13 April 2019, https://www.electronicshub.org/pid-controller-working-and-tuning-methods/.

[11] Rajesh, R.: *Vehicle Dynamics and Control.* Springer, 2012. 526 p. ISBN 1461414326.

[12] Ruedas de Prensa: *Porsche Panamera*, accessed 13 April 2019, https://www.km77.com/coches/porsche/panamera/2017/estandar/informacion/porsche-panamera-2017-impresiones-de-conduccion.

[13] Vandi, G., et al.: Vehicle dynamics modeling for real-time simulations. *In SAE Technical Paper*, September 2013,

[14] Vehico: *ISO Double Lane Change Test*, accessed 13 April 2019, http://www.vehico.com/index.php/en/applications/iso-lane-change-test.

[15] Vlk, F.: *Dynamika motorových vozidel: jízdní odpory: hnací charakteristika: brzdění: odpružení: řiditelnost: ovladatelnost: stabilita.* VLK, 2000. 434 p. ISBN 80-238-5273-6.

List of abbreviations

CAN	Controller area network
CG	Center of gravity
ECU	Electronic control unit
EPS	Electric power steering
GPS	Global position system
LKA	Lane keeping assist
PC	Personal computer
USB	Universal serial bus

List of symbols

a	acceleration
α	tire side-slip angle
β	vehicle side-slip angle
C_α	tire cornering stiffness factor
δ	wheel angle
F_x	longitudinal force
F_y	lateral force
F_z	normal force
I_z	moment of inertia about axis Z
l	distance between rear and front wheel axle
m	vehicle weight
μ	friction coefficient
R	cornering radius
ψ	vehicle yaw
$\dot{\psi}$	vehicle yaw rate

Content of CD

Final programs	Folder with final version of programs that are running on the test vehicle
Literature	Folder with important and useful literature
Measurements	Data from measurements from real vehicle
Simulations	Simulation of vehicle, programs are in Simulink
Thesis	Folder with diploma thesis written in LaTeX.